Identifying Intention Posts in Discussion Forums

Zhiyuan Chen, Bing Liu

Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA
czyuanacm@gmail.com, liub@cs.uic.edu

Meichun Hsu, Malu Castellanos, Riddhiman Ghosh

HP Labs
Palo Alto, CA 94304, USA
{meichun.hsu, malu.castellanos, riddhiman.ghosh}@hp.com

Abstract

This paper proposes to study the problem of identifying intention posts in online discussion forums. For example, in a discussion forum, a user wrote "I plan to buy a camera," which indicates a buying intention. This intention can be easily exploited by advertisers. To the best of our knowledge, there is still no reported study of this problem. Our research found that this problem is particularly suited to transfer learning because in different domains, people express the same intention in similar ways. We then propose a new transfer learning method which, unlike a general transfer learning algorithm, exploits several special characteristics of the problem. Experimental results show that the proposed method outperforms several strong baselines, including supervised learning in the target domain and a recent transfer learning method.

1 Introduction

Social media content is increasingly regarded as an information gold mine. Researchers have studied many problems in social media, e.g., sentiment analysis (Pang & Lee, 2008; Liu, 2010) and social network analysis (Easley & Kleinberg, 2010). In this paper, we study a novel problem which is also of great value, namely, *intention identification*, which aims to identify discussion posts expressing certain user intentions that can be exploited by businesses or other interested parties. For example, one user wrote, "*I am looking for a brand new car to replace my old Ford Focus*". Identifying such intention automatically can help social media sites to decide what ads to display so that the ads are more likely to be clicked.

This work focuses on identifying user posts with *explicit* intentions. By *explicit* we mean that the intention is explicitly stated in the text, no need to deduce (hidden or implicit intention). For example, in the above sentence, the author clearly expressed that he/she wanted to buy a car. On the other hand, an example of an implicit sentence is "*Anyone knows the battery life of iPhone*?" The person may or may not be thinking about buying an iPhone.

To our knowledge, there is no reported study of this problem in the context of text documents. The main related work is in Web search, where *user* (or *query*) *intent classification* is a major issue (Hu et al., 2009; Li, 2010; Li, Wang, & Acero, 2008). Its task is to determine what the user is searching for based on his/her keyword queries (2 to 3 words) and his/her click data. We will discuss this and other related work in Section 2.

We formulate the proposed problem as a two-class classification problem since an application may only be interested in a particular intention. We define *intention posts* (positive class) as the posts that explicitly express a particular intention of interest, e.g., the intention to buy a product. The other posts are *non-intention posts* (negative class). Note that we do not exploit intention specific knowledge since our aim is to propose a generic method applicable to different types of intentions.

There is an important feature about this problem which makes it amenable to transfer learning

so that we do not need to label data in every domain. That is, for a particular kind of intention such as buying, the ways to express the intention in different domains are often very similar. This fact can be exploited to build a classifier based on labeled data in some domains and apply it to a new/target domain without labeling any training data in the target domain. However, this problem also has some special difficulties that existing general transfer learning methods do not deal with. The two special difficulties of the proposed problem are as follows:

1. In an intention post, the intention is typically expressed in only one or two sentences while most sentences do not express intention, which provide very noisy data for classifiers. Furthermore, words/phrases used for expressing intention are quite limited compared to other types of expressions. These mean that the set of shared (or common) features in different domains is very small. Most of the existing advanced transfer learning methods all try to extract and exploit these shared features. The small number of such features in our task makes it hard for the existing methods to find them accurately, which in turn learn poorer classifiers.
2. As mentioned above, in different domains, the ways to express the same intention are often similar. This means that only the positive (intention) features are shared among different domains, while features indicating the negative class in different domains are very diverse. We then have an imbalance problem, i.e., the shared features are almost exclusively features indicating the positive class. To our knowledge, none of the existing transfer learning methods deals with this imbalance problem of shared features, which also results in inaccurate classifiers.

We thus propose a new transfer learning (or domain adaptation) method, called Co-Class, which, unlike a general transfer learning method, is able to deal with these difficulties in solving the problem. Co-Class works as follows: we first build a classifier h using the labeled data from existing domains, called the source data, and then apply the classifier to classify the target (domain) data (which is unlabeled). Based on the target data labeled by h, we perform a feature selection on the target data. The selected set of features is used to build two classifiers, one (h_S) from the labeled source data and one (h_T) from the target data which has been labeled by h. The two classifiers (h_S and h_T) then work together to perform classification of the target data. The process then runs iteratively until the labels assigned to the target data stabilize. Note that in each iteration both classifiers are built using the same set of features selected from the target domain in order to focus on the target domain. The proposed Co-Class explicitly deals with the difficulties mentioned above (see Section 3). Our experiments using four real-life data sets extracted from four forum discussion sites show that Co-Class outperforms several strong baselines. What is also interesting is that it works even better than fully supervised learning in the target domain itself, i.e., using both training and test data in the target domain. It also outperforms a recent state-of-the-art transfer learning method (Tan et al., 2009), which has been successfully applied to the NLP task of sentiment classification.

In summary, this paper makes two main contributions:

1. It proposes to study the novel problem of intention identification. User intention is an important type of information in social media with many applications. To our knowledge, there is still no reported study of this problem.
2. It proposes a new transfer learning method Co-Class which is able to exploit the above two key issues/characteristics of the problem in building cross-domain classifiers. Our experimental results demonstrate its effectiveness.

2 Related Work

Although we have not found any paper studying intention classification of social media posts, there are some related works in the domain of Web search, where user or query intent classification is a major issue (Hu et al., 2009; Li, 2010; Li et al., 2008). The task there is to classify a query submitted to a search engine to determine what the user is searching for. It is different from our problem because they classify based on the user-submitted keyword queries (often 2 to 3 words) together with the user's click-through data (which represent the user's behavior). Such intents are typically implicit because people usually do not issue a search query like "*I want to buy a digital cam-*

era." Instead, they may just type the keywords "*digital camera*". Our interest is to identify *explicit* intents expressed in full text documents (forum posts). Another related problem is online commercial intention (OCI) identification (Dai et al., 2006; Hu et al., 2009), which focuses on capturing commercial intention based on a user query and web browsing history. In this sense, OCI is still a user query intent problem.

In NLP, (Kanayama & Nasukawa, 2008) studied users' needs and wants from opinions. For example, they aimed to identify the user needs from sentences such as "*I'd be happy if it is equipped with a crisp LCD*." This is clearly different from our explicit intention to buy or to use a product/service, e.g., "*I plan to buy a new TV*."

Our proposed Co-Class technique is related to transfer learning or domain adaptation. The proposed method belongs to "*feature representation transfer*" from source domain to target domain (Pan & Yang, 2010). Aue & Gamon (2005) tried training on a mixture of labeled reviews from other domains where such data are available and test on the target domain. This is basically one of our baseline methods 3TR-1TE in Section 4. Their work does not do multiple iterations and does not build two separate classifiers as we do. Some related methods were also proposed in (W. Dai, Xue, Yang & Yu, 2007; Tan et al., 2007; Yang, Si & Callan, 2006). More sophisticated transfer learning methods try to find common features in both the source and target domains and then try to map the differences of the two domains (Blitzer, Dredze, & Pereira, 2007; Pan, et al, 2010; Bollegala, Weir & Carroll, 2011; Tan et al., 2009). Some researchers also used topic modeling of both domains to transfer knowledge (Gao & Li, 2011; He, Lin & Alani, 2011). However, none of these methods deals with the two problems/difficulties of our task. Co-Class tackles them explicitly and effectively (Section 4).

The proposed Co-Class method is also related to Co-Training method in (Blum & Mitchell, 1998). We will compare them in detail in Section 3.3.

3 The Proposed Technique

We now present the proposed technique. Our objective is to perform classification in the target domain by utilizing labeled data from the source domains. We use the term "source domains" as we can combine labeled data from multiple source domains. The target domain has no labeled data. Only the source domain data are labeled.

To deal with the first problem in Section 1 (i.e., the difficulty of finding common features across different domains), Co-Class avoids it by using an EM-based method to iteratively transfer from the source domains to the target domain while exploiting feature selection in the target domain to focus on important features in the target domain.

Since our ideas are developed starting from the EM (Expectation Maximization) algorithm and its shortcomings, we now introduce EM.

3.1 EM Algorithm

EM (Dempster, Laird, & Rubin, 1977) is a popular class of iterative algorithms for maximum likelihood estimation in problems with incomplete data. It is often used to address missing values in the data by computing expected values using existing values. The EM algorithm consists of two steps, the Expectation step (E-step) and the Maximization step (M-step). E-step basically fills in the missing data, and M-step re-estimates the parameters. This process iterates until convergence. Since our target data have no labels, which can be treated as missing values/data, the EM algorithm naturally applies. For text classification, each iteration of EM (Nigam, McCallum, Thrun, & Mitchell, 2000) usually uses the naïve Bayes (NB) classifier. Below, we first introduce the NB classifier.

Given a set of training documents D, each document $d_i \in D$ is an ordered list of words. We use $w_{d_i,k}$ to denote the word in the position k of d_i, where each word is from the vocabulary $V = \{w_1, \ldots, w_{|v|}\}$, which is the set of all words considered in classification. We also have a set of classes $C = \{+, -\}$ representing positive and negative classes. For classification, we compute the posterior probability $\Pr(c_j|d_i)$. Based on the Bayes rule and multinomial model, we have:

$$\Pr(c_j) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|} \quad (1)$$

and with Laplacian smoothing:

$$\Pr(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i) \Pr(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N(w_s, d_i) \Pr(c_j \mid d_i)} \quad (2)$$

where $N(w_t, d_i)$ is the number of times that the word w_t occurs in document d_i, and $\Pr(c_j|d_i) \in [0,1]$ is the probability of assigning class c_j to d_i. Assuming that word probabilities are independent given a class, we have the NB classifier:

$$\Pr(c_j \mid d_i) = \frac{\Pr(c_j)\prod_{k=1}^{|d_i|}\Pr(w_{d_i,k} \mid c_j)}{\sum_{r=1}^{|C|}\Pr(c_r)\prod_{k=1}^{|d_i|}\Pr(w_{d_i,k} \mid c_r)} \quad (3)$$

The EM algorithm basically builds a classifier iteratively using NB and both the labeled source data and the unlabeled target data. However, the major shortcoming is that the feature set, even with feature selection, may fit the labeled source data well but not the target data because the target data has no labels to be used in feature selection. Feature selection is shown to be very important for this application as we will see in Section 4.

3.2 FS-EM

Based on the discussion above, the key to solve the problem of EM is to find a way to reflect the features in the target domain during the iterations. We propose two alternatives, FS-EM (Feature Selection EM) and Co-Class (Co-Classification). This sub-section presents FS-EM.

EM can select features only before iterations using the labeled source data and keep using the same features in each iteration. However, these features only fit the labeled source data but not the target data. We then propose to select features during iterations, i.e., after each iteration, we redo feature selection. For this, we use the predicted classes of the target data. In naïve Bayes, we define the predicted class for document d_i as

$$c = \underset{c_j \in C}{\operatorname{argmax}} \Pr(c_j|d_i) \quad (4)$$

The detailed algorithm for FS-EM is given in Figure 1. First, we select a feature set from the labeled source data D_L and then build an initial NB classifier (lines 1 and 2). The feature selection is based on Information Gain, which will be introduced in Section 3.4. After that, we classify each document in the target data D_U to obtain its predicted class (lines 4-6). A new target data set D_P is produced in line 7, which is D_U with added classes (predicted in line 5). Line 8 selects a new feature set Δ from the data D' (which is discussed below), from which a new classifier h is built (line 9). The iteration stops when the predicted classes of D_U do not change any more (line 10).

We now turn to the data set D', which can be formed with one of the two methods:

1. $D' = D_L \cup D_P$
2. $D' = D_P$

The first method (called FS-EM1) merges the labeled source data D_L and the target data D_P (with predicted classes). However, this method does not work well because the labeled source data can dominate D' and the target domain features are still not well represented.

The second method ($D' = D_P$), denoted as FS-EM2, selects features from the target domain data D_P only based on the predicted classes. The classifiers are built in iterations (lines 3-10) using only the target domain data. The weakness of this is that it completely ignores the labeled source data after initialization, but the source data does contain some valuable information. Our final proposed method Co-Class is able to solve this problem.

3.3 Co-Class

Co-Class is our final proposed algorithm. It considers both the source labeled data and the target data with predicted classes. It uses the idea of FS-EM, but is also inspired by Co-Training in (Blum & Mitchell, 1998). It additionally deals with the second issue identified in Section 1 (i.e., the imbalance of shared positive and negative features).

Co-Training is originally designed for semi-supervised learning to learn from a small labeled and a large unlabeled set of training examples, which assumes the set of features in the data can be partitioned into two subsets, and each subset is sufficient for building an accurate classifier. The proposed Co-Class model is similar to Co-Training in that it also builds two classifiers. However, unlike Co-Training, Co-Class does not partition the feature space. Instead, one classifier is built based on the target data with predicted classes (D_P), and the other classifier is built using only the source labeled data (D_L). Both classifiers use the same features (this is an important point) that are selected from the target data D_P only, in order to focus on the target domain. The final classification is based on both classifiers. Furthermore, Co-Training only uses the data from the

Algorithm FS-EM

Input: Labeled data D_L and unlabeled data D_U

1 Select a feature set Δ based on IG from D_L;
2 Learn an initial naïve Bayes classifier h from D_L based on Δ (using Equations (1) and (2));
3 **repeat**
4 **for** each document d_i in D_U **do**
5 $c = h(d_i)$; // predict the class of d_i using h
6 **end**
7 Produce data D_P based on predicted class of D_U;
8 Select a new feature set Δ from D';
9 Learn a new classifier h on D' based on the new feature set Δ;
10 **until** the predicted classes of D_U stabilize
11 Return the classifier h from the last iteration.

Figure 1 – The FS-EM algorithm

same domain.

The detailed Co-Class algorithm is given in Figure 2. Lines 1-6 are the same as lines 1, 2 and 4-7 in FS-EM. Line 8 selects new features Δ from D_P. Two naïve Bayes classifiers, h_L and h_P, are then built using the source data D_L and predicted target data D_P respectively with the same set of features Δ (lines 9-10). Lines 11-13 classify each target domain document d_i using the two classifiers. $\Phi\big(h_L(d_i), h_P(d_i)\big)$ is the aggregate function to combine the results of two classifiers. It is defined as:

$$\Phi\big(h_L(d_i), h_P(d_i)\big) = \begin{cases} + & h_L(d_i) = h_P(d_i) = + \\ - & Otherwise \end{cases}$$

This aims to deal with the imbalanced feature problem. As discussed before, the expressions for stating a particular intention (e.g., buying) are very similar across domains but the non-intention expressions across domains are highly diverse, which result in strong positive features and weak negative features. We then need to restrict the positive class by requiring both classifiers to give positive predictions. If we use the method in Co-Training (multiplying the probabilities of the two NB classifiers), the classification results deteriorate from iteration to iteration because the positive class recall gets higher and higher due to strong positive features, but the precision gets lower and lower.

Since we build and use two classifiers for the final classification, we call the method *Co-Class*, short for *Co-Classification*. Co-Class is different from EM (Nigam et al., 2000) in two main aspects.

Algorithm Co-Class

Input: Labeled data D_L and unlabeled data D_U

1 Select a feature set Δ based on IG from D_L;
2 Learn an initial naïve Bayes classifier h from D_L based on Δ (using Equations (1) and (2));
3 **for** each document d_i in D_U **do**
4 $c = h(d_i)$; // predict the class of d_i using h
5 **end**
6 Produce data D_P based on the predicted class of D_U;
7 **repeat**
8 Select a new feature set Δ from D_P;
9 Build a naïve Bayes classifier h_L using Δ and D_L;
10 Build a naïve Bayes classifier h_P using Δ and D_P;
11 **for** each document d_i in D_U **do**
12 $c_i = \Phi\big(h_L(d_i), h_P(d_i)\big)$; // Aggregate function
13 **end**
14 Produce data D_P based on predicted class of D_U;
15 **until** the prediction classes of D_U stabilize
16 Return classifiers h_L and h_P from the last iteration.

Figure 2 – The Co-Class algorithm

First, it integrates feature selection into the iterations, which has not been done before. Feature selection refines features to enhance the correlation between the features and classes. Second, two classifiers are built based on different domains and combined to improve the classification. Only one classifier is built in existing EM methods, which gives poorer results (Section 4).

3.4 Feature Selection

As feature selection is important for our task, we briefly introduce the Information Gain (IG) method given in (Yang & Pedersen, 1997), which is a popular feature selection algorithm for text classification. IG is based on entropy reflecting the purity of the categories or classes by knowing the presence or absence of each feature, which is defined as:

$$IG(f) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + \sum_{f,\bar{f}} P(f) \sum_{i=1}^{m} P(c_i \mid f)\log P(c_i \mid f)$$

Using the IG value of each feature f, all features can be ranked. As in normal classification tasks, the common practice is to use a set of top ranked features for classification.

4 Evaluation

We have conducted a comprehensive set of experiments to compare the proposed Co-Class method with several strong baselines, including a state-of-

the-art transfer learning method.

4.1 Experiment Settings

Datasets: We created 4 different domain datasets crawled from 4 different forum discussion sites:

Cellphone: http://www.howardforums.com/forums.php
Electronics: http://www.avsforum.com/avs-vb/
Camera: http://forum.digitalcamerareview.com/
TV: http://www.avforums.com/forums/tvs/

For our experiments, we are interested in the intention to *buy*, which is our intention or positive class. For each dataset, we manually labeled 1000 posts.

Labeling: We initially labeled about one fifth of posts by two human annotators. We found their labels highly agreed. We then used only one annotator to complete the remaining labeling. The reason for the strong labeling agreement is that we are interested in only explicit buying intentions, which are clearly expressed in each post, e.g., "*I am in the market for a new smartphone*." There is little ambiguity or subjectivity in labeling.

To ensure that the task is realistic, for all datasets we keep their original class distributions as they are extracted from their respective websites to reflect the real-life situation. The intention class is always the minority class, which makes it much harder to predict due to the imbalanced class distribution. Table 1 gives the statistics of each dataset. On average, each post contains about 7.5 sentences and 122 words. We have made the datasets used in this paper publically available at the websites of the first two authors.

Evaluation measures: For all experiments, we use *precision*, *recall* and *F1-score* as the evaluation measures. They are suitable because our objective is to identify intention posts.

4.2 One Domain Learning

The objective of our work is to classify the target domain instances without labeling any target domain data. To set the background, we first give the results of one domain learning, i.e., assuming that there is labeled training data in the target domain (which is the traditional fully supervised learning). We want to see how the results of Co-Class compare with the fully supervised learning.

Dataset	No. of Intention	No. of Non-Intention	Total No. of posts
Cellphone	184	816	1000
Electronics	280	720	1000
Camera	282	718	1000
TV	263	737	1000

Table 1: Datasets statistics with the buy intention

For this set of experiments, we use naïve Bayes and SVM. For naïve Bayes, we use the *Lingpipe* implementation (http://alias-i.com/lingpipe/). For SVM, we use SVM^{Light} (Joachims, 1999) from (http://svmlight.joachims.org/) with the linear kernel as it has been shown by many researchers that linear kernel is sufficient for text classification (Joachims, 1998; Yang and Liu, 1999).

During labeling, we observed that the intention in an intention (positive) post is often expressed in the first few or the last few sentences. Hence, we tried to use the full post (denoted by Full), the first 5 sentences (denoted by (5, 0)), and first 5 and last 5 sentences (denoted by (5, 5)). We also experimented with the first 3 sentences, and first 3 and last 3 sentences but their results were poorer.

The experiments were done using 10-fold cross validation. For the number of selected features, we tried 500, 1000, 1500, 2000, 2500 and all. We also tried unigrams, bigrams, trigrams, and 4-grams. To compare naïve Bayes with SVM, we tried each combination, i.e. number of features and n-grams, and found the best model for each method. We found that naïve Bayes works best when using trigrams with 1500 selected features. Bigrams with 1000 features are the best combination for SVM. Figure 3 shows the comparison of the best results (F1-scores) of naïve Bayes and SVM.

From Figure 3, we make the following observations:

1. SVM does not do well for this task. We tuned the parameters of SVM, but the results were similar to the default setting, and all were

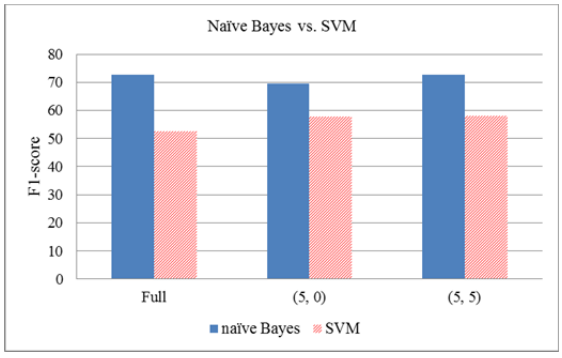

Figure 3 – Naïve Bayes vs. SVM

Naïve Bayes (n-grams, features)	Cellphone			Electronics			Camera			TV		
	Full	**5,0**	**5,5**	**Full**	**5,0**	**5,5**	**Full**	**5,0**	**5,5**	**Full**	**5,0**	**5,5**
Unigrams, 2000	59.91	55.21	56.76	71.31	70.10	71.24	71.57	71.53	75.78	74.96	74.45	74.13
Bigrams, 1500	61.97	54.29	59.17	70.71	71.46	72.48	77.02	74.12	77.38	79.76	77.71	79.72
Trigrams, 1500	**61.50**	**55.78**	**60.15**	**71.38**	**71.07**	**71.61**	**77.66**	**75.71**	**78.74**	**80.24**	**75.66**	**79.92**
4-grams, 2000	58.94	51.94	57.72	72.03	71.98	73.05	79.84	75.09	79.46	79.12	76.61	79.88

Table 2: One-domain learning using naïve Bayes with n-grams (with best no. of features)

Naïve Bayes (n-grams, features)	Cellphone			Electronics			Camera			TV		
	Full	**5,0**	**5,5**	**Full**	**5,0**	**5,5**	**Full**	**5,0**	**5,5**	**Full**	**5,0**	**5,5**
Trigrams, 2000	57.98	57.60	58.67	71.85	69.74	71.51	74.45	73.58	74.24	74.07	71.34	73.65
Trigrams, 2500	58.08	57.48	59.12	72.27	69.65	71.82	76.15	73.64	76.31	74.02	71.25	73.49
Trigrams, 3000	**56.74**	**56.94**	**56.74**	**72.27**	**70.76**	**72.43**	**77.62**	**74.65**	**77.62**	**75.64**	**71.65**	**74.73**
Trigrams, 3500	56.60	56.81	57.21	71.86	70.40	72.24	77.17	74.85	76.68	74.25	71.10	73.37
4-grams, 2000	58.94	51.94	57.72	72.03	71.98	73.05	79.84	75.09	79.46	79.12	76.61	79.88

Table 3: F1-scores of 3TR-1TE with trigrams and different no. of features

worse than naïve Bayes. We believe the main reason is that the data for this application is highly noisy because apart from one or two intention sentences, other sentences in an intention post have little difference from those in a non-intention post. SVM does not perform well with very noisy data. When there are data points far away from their own classes, SVM tends to be strongly affected by such points (Wu & Liu, 2007). Naïve Bayes is more robust in the presence of noise due to its probabilistic nature.

2. SVM using only the first few and/or last few sentences performs better than using full posts because full posts have more noise. However, it is still worse than naïve Bayes.
3. For naïve Bayes, using full posts and the first 5 and last 5 (5, 5) sentences give similar results, which is not surprising as (5, 5) has almost all the information needed. Without using the last 5 sentence (5, 0), the results are poorer.

We also found that without feature selection (using all features), the results are markedly worse for both naïve Bayes and SVM. This is understandable (as we discussed earlier) because most words and sentences in both intention and non-intention posts are very similar. Thus, feature selection is highly desirable for this application.

Effect of different combinations: Table 2 gives the detailed F1-score results of naïve Bayes with best results in different n-grams (with best number of features). We can see that using trigrams produces the best results on average, but bigrams and 4-grams are quite similar. It turns out that using trigrams with 1500 selected features performs the best. SVM results are not shown as they are poorer.

In summary, we say that naïve Bayes is more suitable than SVM for our application and feature selection is crucial. In our experiments reported below, we will only use naïve Bayes with feature selection.

4.3 Evaluation of Co-Class

We now compare Co-Class with the baseline methods listed below. Note that for this set of experiments, the source data all contain labeled posts from three domains and the target data contain unlabeled posts in one domain. That is, for each target domain, we merge three other domains for training and the target domain for testing. For example, for the target of "Cellphone", the model is built using the data from the other three domains (i.e., "Electronics", "Camera" and "TV"). The results are the classification of the model on the target domain "Cellphone". Several strong baselines are described as follows:

3TR-1TE: Use labeled data from three domains to train and then classify the target (test) domain. There is no iteration. This method was used in (Aue & Gamon, 2005).

EM: This is the algorithm in Section 3.1. The combined data from three domains are used as the labeled source data. The data of the remaining one domain are used as the unlabeled target data, which is also used as the test data (since it is unlabeled).

ANB: This is a recent transfer learning method

(Tan et al., 2009). ANB uses *frequently co-occurring entropy* (FCE) to pick out generalizable (or shared) features that occur frequently in both the source and target domains. Then, a weighted transfer version of naïve Bayes classifier is applied. We chose this method for comparison as it is a recent method, also based on naïve Bayes, and has been applied to the NLP task of sentiment classification, which to some extend is related to the proposed task of intention classification. ANB was also shown to perform better than EM and naïve Bayes transfer learning method (Dai et al., 2007).

We look at the results of 3TR-1TE first, which are shown in Table 3. Due to space limitations, we only show the trigrams F1-scores as they perform the best on average. Table 3 gives the number of features with trigrams. We can observe that on average using 3000 features gives the best F1-score results. It has 1000 more features than one domain learning because we now combine three domains (3000 posts) for training and thus more useful features.

From Table 3, we observe that the F1-score results of 3TR-1TE are worse than those of one domain learning (Table 2), which is intuitive because no training data are used from the target domain. But the results are not dramatically worse which indicate that there are some common features in different domains, meaning people expressing the same intention in similar ways.

Since we found that trigrams with 3000 features perform the best on average, we run EM, FS-EM1, FS-EM2 and Co-Class based on trigrams with 3000 features. For the baseline ANB, we tuned the parameters using a development set (1/10 of the training data). We found that selecting 2000 generalizable/shared features gives the best results (the default is 500 in (Tan et al., 2009)). We kept ANB's other original parameter values. The F1-scores (averages over all 4 datasets) with the number of iterations are shown in Figure 4. Iteration 0 is the result of 3TR-1TE. From Figure 4, we can make the following observations:

1. EM makes a little improvement in iteration 1. After that, the results deteriorate. The gain of iteration 1 shows that incorporating the target domain data (unlabeled) is helpful. However, the selected features from source domains can only fit the labeled source data but not the target data, which was explained in Section 3.1.
2. ANB improves slightly from iteration 1 to iteration 6, but the results are all worse than those of Co-Class. We checked the generalizable/shared features of ANB and found that they were not suitable for our problem since they were mainly adjectives, nouns and sentiment verbs, which do not have strong correlation with intentions. This shows that it is hard to find the truly shared features indicating intentions. Furthermore, ANB's results are almost the same as those of EM.
3. FS-EM2 behaves similarly to FS-EM1. After two iterations, the results start to deteriorate. Selecting features only from the target domain makes sense since it can reflect target domain data well. However, it also becomes worse with the increased number of iterations, due to strong positive features. With increased iterations, positive features get stronger due to the imbalanced feature problem discussed in Section 1.
4. Co-Class performs much better than all other methods. With the increased number of iterations, the results actually improve. Starting from iteration 7, the results stabilize. Co-Class solves the problem of strong positive features by requiring strong conditions for positive classification and focusing on features in the target domain only. Although the detailed results of precision and recall are not shown, the Co-Class model actually improves the F1-score by improving both the precision and recall.

Significance of improvement: We now discuss the significance of improvements by comparing the results of Co-Class with other models. Table 4 summarizes the results among the models. For Co-Class, we use the converged models at iteration 7. We also include the One Domain learning results which are from fully supervised classification in the target domains with trigrams and 1500 features. The results of 3TR-1TE, EM, ANB, FS-EM1, and FS-EM2 are obtained based on their settings which give the best results in Figure 4.

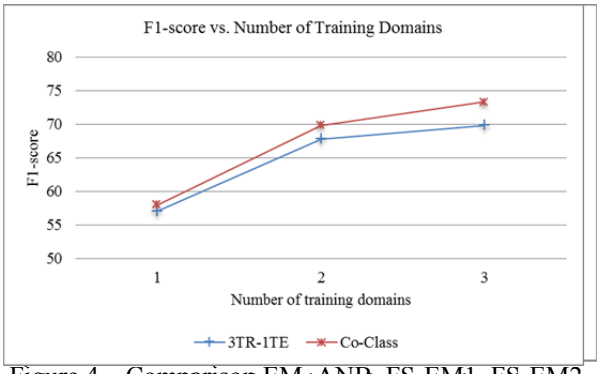

Figure 4 – Comparison EM, ANB, FS-EM1, FS-EM2, and Co-Class across iterations (0 is 3TR-1TE)

Figure 5 – Effect of number of source domains using 3TR-1TE and Co-Class.

It is clear from Table 4 that Co-Class is the best method in general. It is even better than the fully supervised One-Domain learning, although their results are not strictly comparable because One-Domain learning uses training and test data from the same domain via 10-fold cross validation, while all other methods use one domain as the test data (the labeled data are from the other three domains). One possible reason is that the labeled data are much bigger than those in One-Domain learning, which contain more expressions of buying intention. Note that FS-EM1 and FS-EM2 work slightly better than Co-Class in domain "Camera" because it is the least noisy domain with very short posts while other domains (as source data) are quite noisy. With good quality data, FS-EM1 and FS-EM2 (also proposed in this paper) can do slightly better than Co-Class. Statistical paired *t*-test shows that Co-Class performs significantly better than baseline methods 3TR-1TE, EM, ANB and FS-EM1 at the confidence level of 95%, and better than FS-EM2 at the confidence level of 94%.

Effect of the number of training domains: In our experiments above, we used 3 source domain data and tested on one target domain. We now show what happens if we use only one or two source domain data and test on one target domain. We tried all possible combinations of source and target data. Figure 5 gives the average results over the four target/test domains. We can see that using more source domains is better due to more labeled data. With more domains, Co-Class also improves more over 3TR-1TE.

5 Conclusion

This paper studied the problem of identifying intention posts in discussion forums. The problem has not been studied in the social media context. Due to special characteristics of the problem, we found that it is particularly suited to transfer learning. A new transfer learning method, called Co-Class, was proposed to solve the problem. Unlike a general transfer learning method, Co-Class can deal with two specific difficulties of the problem to produce more accurate classifiers. Our experimental results show that Co-Class outperforms strong baselines including classifiers trained using labeled data in the target domains and classifiers from a state-of-the-art transfer learning method.

Acknowledgments

This work was supported in part by a grant from National Science Foundation (NSF) under grant no. IIS-1111092, and a grant from HP Labs Innovation Research Program.

References

Aue, A., & Gamon, M. (2005). Customizing Sentiment Classifiers to New Domains: A Case Study. *Proceedings of Recent Advances in Natural Language Processing* (*RANLP*).

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL*).

Model	Cellphone			Electronics			Camera			TV		
	Full	**5,0**	**5,5**	**Full**	**5,0**	**5,5**	**Full**	**5,0**	**5,5**	**Full**	**5,0**	**5,5**
One-Domain	61.50	55.78	60.15	71.38	71.07	71.61	77.66	75.71	78.74	80.24	75.66	79.92
3TR-1TE	56.74	56.94	56.74	72.27	70.76	72.43	77.62	74.65	77.62	75.64	71.65	74.73
EM	60.28	59.59	60.45	70.47	69.90	71.33	79.38	77.01	80.31	74.96	70.76	74.31
ANB	62.53	59.29	62.41	66.58	68.29	68.36	78.37	77.49	78.83	78.70	75.73	78.26
FS-EM1	59.01	57.69	59.41	70.75	71.74	72.00	80.58	76.13	80.37	79.29	73.75	77.34
FS-EM2	59.54	60.09	61.33	71.19	72.09	72.07	80.14	77.93	81.09	78.90	74.21	77.53
Co-Class	**62.69**	**61.10**	**62.69**	**73.38**	**73.23**	**73.95**	**79.69**	**74.65**	**78.66**	**81.12**	**76.40**	**81.60**

Table 4: F1-score results of One-Domain, 3TR-1TE, EM, ANB, FS-EM1, FS-EM2, and Co-Class

Blum, A., & Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. COLT: *Proceedings of the eleventh annual conference on Computational learning theory*.

Bollegala, D., Weir, D. J., & Carroll, J. (2011). Using Multiple Sources to Construct a Sentiment Sensitive Thesaurus for Cross-Domain Sentiment Classification. *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL*).

Dai, H. K., Zhao, L., Nie, Z., Wen, J. R., Wang, L., & Li, Y. (2006). Detecting online commercial intention (OCI). *Proceedings of the 15th international conference on World Wide Web* (*WWW*).

Dai, W., Xue, G., Yang, Q., & Yu, Y. (2007). Transferring naive bayes classifiers for text classification. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence* (*AAAI*).

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*. Series B, 39(1), 1–38.

Easley, D., & Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press.

Gao, S., & Li, H. (2011). A cross-domain adaptation method for sentiment classification using probabilistic latent analysis. *Proceedings of the 20th ACM international conference on Information and knowledge management* (*CIKM*).

He, Y., Lin, C., & Alani, H. (2011). Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (*ACL*).

Hu, D. H., Shen, D., Sun, J.-T., Yang, Q., & Chen, Z. (2009). Context-Aware Online Commercial Intention Detection. *Proceedings of the 1st Asian Conference on Machine Learning: Advances in Machine Learning* (*ACML*).

Hu, J., Wang, G., Lochovsky, F., tao Sun, J., & Chen, Z. (2009). Understanding user's query intent with wikipedia. *Proceedings of the 18th international conference on World wide web* (*WWW*).

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *European Conference on Machine Learning* (*ECML*).

Joachims, T. (1999). Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Kanayama, H., & Nasukawa, T. (2008). Textual Demand Analysis: Detection of Users' Wants and Needs from Opinions. *Proceedings of the 22nd International Conference on Computational Linguistics* (*COLING*).

Li, X. (2010). Understanding the Semantic Structure of Noun Phrase Queries. *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL*).

Li, X., Wang, Y.-Y., & Acero, A. (2008). Learning query intent from regularized click graphs. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (*SIGIR*).

Liu, B. (2010). Sentiment Analysis and Subjectivity. (N. Indurkhya & F. J. Damerau, Eds.) *Handbook of Natural Language Processing*, 2nd ed.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Mach. Learn.*, 39(2-3), 103–134.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., & Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. *Proceedings of the 19th international conference on World wide web* (*WWW*).

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.*, 22(10), 1345–1359.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 1–135.

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval* (*ECIR*).

Tan, S., Wu, G., Tang, H., & Cheng, X. (2007). A novel scheme for domain-transfer problem in the context of sentiment analysis. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (*CIKM*).

Wu, Y., & Liu, Y. (2007). Robust truncated-hinge-loss support vector machines. *Journal of the American Statistical Association*, 102(479), 974–983.

Yang, H., Si, L., & Callan, J. (2006). Knowledge Transfer and Opinion Detection in the TREC 2006 Blog Track. *Proceedings of TREC*.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (*SIGIR*).

Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (*ICML*).