Opinion Detection by Transfer Learning

11-742 Information Retrieval Lab Report

Hui Yang
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA, USA, 15213
huiyang@cs.cmu.edu

ABSTRACT

Opinion detection is the main task of TREC 2006 Blog track, which identifies opinions from text documents in the TREC blog corpus. Given that it is the first year of the task, there is no available training data provided. Using knowledge about how people give opinions on other domains, for example, movie review, product review and book review, is the best available training data for opinion detection in blog domain. This work describes how to apply transfer learning in opinion detection. A Bayesian logistic regression framework is used and knowledge from training data in other domains is captured by a non-diagonal prior covariance matrix. The experimental results show that the approach is effective and achieve an improvement of 32% from baseline.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Opinion Detection

Keywords

Opinion Detection, Transfer Learning

1. INTRODUCTION

Opinion detection is an emerging topic that attracts more and more research interests from researchers in data mining and natural language processing [1][3][5][12][14]. Given a document, opinion detection task identifies and extracts the opinionated expressions for a certain topic. Some opinions expressed in a general way as in "I really like this work", hence words with sentiment polarity are playing an important role to recognizing the presence of an opinion. On the other hand, there are many opinions have its own way to express, for example, "Watching the film is like reading a times portrait of grief that keeps shifting focus to the journalist who wrote it". Given the great variety and complexity of human language, opinion detection is a challenging job.

In year 2006, Text REtrieval Conference (TREC) started a new track to study research topics in the blog domain, and opinion detection in blogs is the main task [7]. Since it is the first year and blog data is pretty new in the research community, there is a lack of training data. Given the lack of training data from blog corpus, simple supervised learning is not possible. How to transfer knowledge about opinions from other domains, which have labelled training data, is another challenge.

This paper gives a try to use techniques in transfer learning [2][8][9][10][11][13] to incorporate common features for opinion detection across different domains to solve the problem of no training data. Bayesian Logistic Regression is the main framework used. The common knowledge is formed into a non-diagonal covariance matrix for the prior of regression coefficients. The learned prior from movie and product reviews is used to estimate whether a sentence is an opinion or not in the blog domain. Moreover, different from classic text classification task, opinion detection has its own effective features in the classification process. This paper also describes "Target-Opinion" word pairs and word synonyms and their effects on opinion detection.

The remainder of this paper is organized into several sections. Section 2 gives a brief literature review of transfer learning, opinion detection and explains the existing work done during TREC 2006. Section 3 details the transfer learning algorithm used in the opinion detection process. Section 4 explains feature selection for opinion detection. Section 5 describes the datasets used in this research. Section 6 elaborates the evaluation and experimental results, also gives an analysis to the results. Section 7 concludes the paper. Appedix A lists the query topics evaluated in this research.

2. RELATED WORK

2.1 Opinion Detection

Researchers in Natural Language Processing (NLP) community are the pioneers for the opinion detection task. Turney [14] groups online words whose point mutual information is close to two words - "excellent" and "poor", and then use them to detect opinions and sentiment polarity. Riloff and Wiebe [5] use a high-precision classifier to get high quality

opinion and non-opinion sentences, and then extract surface text patterns from those sentences to find more opinions and non-opinions and repeat this process to bootstrap. Pang et al. [1] treated opinion and sentiment detection and as a text classification problem and use classical classification methods, like Naive Bayes, Maximum Entropy, Support Vector Machines, with word unigram to predict them. Pang and Lee [3] in their another work also tried to use Minicuts to cluster sentences based on their subjectivity and sentiment orientation. Researchers from data mining community also study the problem of opinion mining. Morinaga et al. [12] used word polarity, syntactic pattern matching rules to extract opinions. They also use principal component analysis to create correspondence between the product names and keywords with the distance on a map showing the closeness.

We participated in TREC-2006 Blog track evaluation. The main task is opinion detection in blog domain. The system [6] is mainly divided into two parts: passage retrieval and opinion classification. During passage retrieval, the topics provided by NIST are parsed and query expansion is done before sending the topics as queries to the Lemur search engine[1]. Documents in the corpus are segmented into passages around 100 words and are the retrieval units for the search engine. The top 5,000 passages returned by Lemur are then sent into a binary text classification program to classified into opinions and non-opinions based the average over their sentence-level subjectivity score. The performance of the system is among top five participated groups.

2.2 Transfer Learning

Transfer learning is to learn from other related tasks and apply the learned model into the current task. The most general form of transfer learning is to learning the similar tasks from one domain to another domain so that transfer the "knowledge" from one to another. In the early research of transfer learning, Baxter [2] and Thrun [13] both used hierarchical Bayesian learning methods to tackle this problem. In recent years, Lawrence and Platt [9], Yu et al. [8] also used hierarchical Bayesian models to learn hyper-parameters of Gaussian process.

Ando and Zhang [10] proposed a framework for Gaussian logistic regression with transfer learning for the task of classification and also provided a theoretical prove for transfer learning in this setting. They learned from multiple tasks to form a good classifier and apply it onto other similar tasks. Raina et al. [11] continued this approach and built informative priors for gaussian logistic regression. These informative priors actually corresponds to the hyper-parameter in other approaches. We follow closely with Raina et al.'s approach and adapt it into the opinion detection task.

3. THE ALGORITHM

After retrieving 5000 paragraphs for each topic, sentence segmentation is done for each paragraph. Though in TREC assessment, document is the evaluation unit, sentence, is actually a more nature unit for the task of opinion detection because different opinions could be present in the same document but much less possible to be present in the same sentence. Therefore, sentence is selected as the basic unit for an opinion in this research.

The remaining task is to identify which sentences contain opinion, which are not. It can be considered as a binary classification problem. Baysian Logistic Regress is the framework used here. Each sentence is represented as $X = [x_1, x_2, ..., x_n]$, where n is the total number of word features x_i. The entire dataset is represented $\mathbf{X} = \{X^{(1)}, X^{(2)}, ..., X^{(m)}\}$, where m is the total number of sentences. A class label for a sentence is either *opinion* or *non-opinion*, and is represented by $Y = \{0, 1\}$.

Logistic regression assumes sigmoid-like data distribution and predicts the class label according to the following formula:

$$P(Y = 1|X = x, \theta) = \frac{1}{1 + \exp(-\theta^T x)} \quad (1)$$

where θ is the regression coefficient. It usually is learned by coordinate descent, while a global optimum is guaranteed to be found.

However, logistic regression, like many other classification /regression algorithms, suffers from overfitting. Usually when large regression coefficients are observed, prediction accuracy is very sensitive to test data, and overfitting occurs. To avoid this problem, usually a multivariate Gaussian prior is added on θ. For simplicity, zero mean and equal variance are assumed. Hence the prior is $N(0, \sigma^2 I)$ and the objective function (in log space) becomes:

$$l(Y = 1|X; \lambda) = \sum_{i=1}^{N} [y_i f(x_i) - log(1 + exp(f(x_i)))] - \frac{1}{2}\lambda \int \{f''(t)\}^2 dt \quad (2)$$

where $f(x) = \theta^T x$. and Maximum A Posteriori (MAP) estimation is:

$$\theta_{map} = \arg\max_{\theta}(\theta^* - \frac{1}{2\sigma^2}||\theta||^2) \quad (3)$$

where θ^* is the maximum likelihood estimation for θ.

The above prior is the most common prior which used in many research problems. It assumes equal variances for all the features, which is not valid in the real world settings. Hence, a general prior with non-diagonal covariance $N(0, \Sigma)$ is used in this research. The MAP estimation becomes:

$$\theta_{map} = \arg\max_{\theta} \sum_i \sum_j (\theta_i^* - \frac{1}{2cov(\theta_i, \theta_j)}\theta_i\theta_j) \quad (4)$$

To apply the above formula, it is required to get the value of $cov(\theta_i, \theta_j)$ for every pair of regression coefficients (θ_i, θ_j). By the definition of covariance, it is the difference of expected joint probability of $E[\theta_i\theta_j]$ and the product of individual expected probability $E[\theta_i]$ and $E[\theta_j]$. The following equation shows the definition of covariance:

$$cov(\theta_i, \theta_j) = E[\theta_i\theta_j] - E[\theta_i]E[\theta_j] \quad (5)$$

Given that the prior's mean is 0, both of individual expected values of θ_i equal to 0, i.e.,$E[\theta_i] = E[\theta_j] = 0$. Therefore, the

[1]Lemur:http://www.lemurproject.org/

covariance of any two regression coefficients becomes:

$$cov(\theta_i, \theta_j) = E[\theta_i \theta_j] \tag{6}$$

which is just the expected joint probability of those two coefficients.

3.1 MCMC for Covariance of Pair-wised Coefficients

The covariance for pair-wised regression coefficients can be obtained by Markov Chain Monte Carlo (MCMC) method. Instead of real covariance, which is not going to be achieved but can be closely estimated by the sample covariance. MCMC suggests to sample several small vocabularies with the two words corresponding to θ_i and θ_j. Each small vocabulary is used as training data to train an ordinary logistic regression model whose objective function is defined in equation 2. The sample covariance is obtained by going through words in each training set and vocabulary.

sample covariance(θ_i, θ_j) =

$$\frac{1}{VT} \sum_{v,t} \theta_i^{(v,t)} \theta_j^{(v,t)} \tag{7}$$

where V is the number of vocabularies and T is the number of training sets from each vocabulary.

Hence the covariance is due to both randomness of vocabularies and training sets. However, only the covariance due to vocabulary change is desired in our case. Hence a correction step is performed through minus a bootstrap estimation of the covariance due to randomness of training set change.

$\text{cov}(\theta_i, \theta_j)$ = sample covariance(θ_i, θ_j) -

$$\frac{1}{V} \sum_{v} \frac{1}{T} \sum_{t} (\theta_i^{(v,t)} - \bar{\theta}_i^{(v)})(\theta_j^{(v,t)} - \bar{\theta}_j^{(v)}) \tag{8}$$

where $\bar{\theta}_i^{(v)}$ and $\bar{\theta}_j^{(v)}$ are sample mean of regression coefficients for each vocabulary across different training sets.

By doing the above calculation, the covariances of each pair of regression coefficient is able to be obtained. However, given that the number of regression coefficients is corresponding to the number of word features, the total amount of computation is huge and not feasible. Therefore, a smarter way of calculating just a small amount of pair-wise covariances is necessary.Moreover, individual pair-wise covariances can only be used to estimate relationship between two words, however, what is needed is to estimate relationship among all the words. In another word, a covariance matrix is the final target to learn.

3.2 Learning a Covariance Matrix

As pointed out in the previous section, it is extremely inefficient to calculate every pair of individual covariances for all word features. Instead, learning indirect common features and representing the word features as those features will dramatically reduce the amount of computations. In this way, only a small fraction of word pairs need to be calculated their pair-wise covariances. And the rest of word pairs' covariances can be estimated by a transformation from their indirect features. Therefore, the problem of learning individual covariance for each word pair is turned into the problem of learning the correspondence between an underline common feature, which will be shared by many word pairs, and a word pair itself. Mathematically, if the indirect common features are defined as a feature vector F_{ij}, and the small fraction of covariances are defined as C, in which all the values are calculated by the method given in section 3.1 and are represented by c_{ij}, the objective function to learn the correspondence ψ is given in the following least squared error function:

$$\min_{\psi} \sum_{(i,j) \in K} (c_{ij} - \psi^T F_{ij})^2 \tag{9}$$

where K is the set of words whose covariances are calculated explicitly.

By learning the correspondence of the word feature and indirect common features, i.e., by learning ψ, the entire covariance matrix C can be estimated by computing its (i, j)th element as :

$$c_{ij} = \psi^T F_{ij} \tag{10}$$

A valid covariance matrix needs to be positive semi-definite (PSD), which is a Hermitian matrix with all of its eigenvalues nonnegative. In other words, it needs to be a square, self-adjoint matrix with nonnegative eigenvalues. Clearly, the individual pair-wise covariances obtained in section 3.1 are not going to be such a matrix automatically. And the covariance matrix obtained by equation 10 is not PSD either. Hence, a projection from the original covariances to a PSD cone is necessary to make the matrix usable. Therefore, the covariance matrix C should be as close to a PSD matrix Σ as possible, which is represented in the following mean squared error objective function:

$$\min_{\Sigma} \sum_{i,j} (c_{ij} - \Sigma_{ij})^2 \tag{11}$$

This can be related to the indirect common features by substituting c_{ij} with $\psi^T F_{ij}$, and the objective finction for getting a PSD matrix becomes:

$$\min_{\Sigma} \sum_{i,j} (\psi^T F_{ij} - \Sigma_{ij})^2 \tag{12}$$

note that different from in equation 10, where ψ is the target to be learned, ψ is a fixed values vector now.

As we can see so far, for each concern of how to learn a good covariance matrix, an objective function is found. To solve the first and second in sequence is less effective and less efficient than solve them as a combined objective function since at the first step, the learned covariance matrix C can be highly indefinite, and hence at the second step, many entries need to be adjusted to satisfying the PSD constraints, and the knowledge learned in the first step is wasted and has to learned again. By combining two objective functions into one, while learning ψ, the consideration of PSD constraints is also effective. Therefore, the overall objective function becomes a joint optimization problem and can be represented as:

$$\min_{\psi, \Sigma} \lambda \sum_{(i,j) \in K} (c_{ij} - \psi^T F_{ij})^2 + (1 - \lambda) \sum_{i,j} (\Sigma_{ij} - \psi^T F_{ij})^2 \tag{13}$$

where λ is the trade-off coefficient between the two sub objectives. As λ goes to 0, only the PSD constraints are taken care of, and as λ goes to 1, only the word pair relationship constraints are taken care of. We set $\lambda = 0.6$ in this research, which is a good trade-off coefficient learned empirically.

The joint optimization problem in equation 13 can be solved in an minimization-minimization procedure by fixing one argument and minimizing on another. In our case, alternatively, ψ is minimized over when Σ is fixed, and Σ is minimized over when ψ is fixed. When minimizing over ψ, quadratic programming (QP) is sufficient. There are many QP sovlers[2] available and can be easiliy obtained. When minimizing over Σ, this is a special semi-definite problem (SDP), and can be easily done by performing eigen-decomposition and keeping the nonnegative eigenvalues, which can be done in any standard SDP solvers.

Since equation 13 is convex, which can be proved, there is a global minimum existing. Therefore, the minimization-minimization procedure repeats the two minimization steps and continues until a guaranteed convergence.

4. FEATURE DESIGN

Given that there is no training data available in the target domain, transfer learning is the only choice besides manually tagging a corpus. The most naive way of transfer learning will be training a model on some external domain's data, which is handy, and using the external domain's vocabulary, creating unigram or bi-gram features, testing on the test corpus, with the hope that some unigram and bi-gram features are also present in the test corpus. Since different features play different roles in different domains, for example, "movie" is a key word feature and appearing in many opinion sentences within the movie review domain, while it is definitely not a key feature for opinion detection in product review, since it has low probability that a sentence talking about movie is an opinion about some product, for example, Canon camera. There is obvious bias between the training set and test set and hence it will not result a very good opinion detection rate. However, this is the baseline transfer learning used in our experiments since it is the simplest way of doing transfer learning.

Another straightforward way of doing transfer learning is to also using word features from other domains, but, instead using word features from just a single domain, using common word features appearing in multiple external domains. The purpose is to find word features which is common enough to appear in every opinion related corpus. For example, in both movie reviews and product reviews, "good", "I like" will indicate a positive opinion, and "disappointed", "hate" will indicate a negative opinion. If only these common "opinion"-related features are extracted and kept in the vocabulary, the severe bias existing in the above approach is resolved. This is the approach that we used in our submission to TREC 2006 Blog track [6] and will be one of the experiment option as well in later section.

The approach used in this paper is to get a common prior, which carries the common knowledge embedded in different opinion related corpora, for logistic regression coefficients. The prior is represented as a Gaussian distribution with non-diagonal covariance, which can be used to represent word to word relationship which is absent in the above two approaches, which treat each word features are identically independent distributed (i.i.d.). The third approach is described in section 3, which forms the word to word relationship as a function of indirect common features across different opinion related corpora. What are the good indirect features for opinion detection is investigated.

One prominent phenomenon in opinion and also one of the difficult part of opinion detection is that people are not always using "blah blah is good", "awesome blah blah!" to express opinions, instead, different opinion targets relate to their own customary opinion expressions. For example, we usually say "A person is knowledgeable" and "A computer processor is fast", not "A person is fast" and "A computer processor is knowledgable". Target-specific opinions are not to be well-identified with simple word polarity test either. For example, "A computer processor is running like a horse". There is no positive or negative adjectives available in the sentence and polarity test will say this is not an opinion even though it is one in fact.

To model the correspondence of a target and its customary opinion expression, a feature, which is a pair of (target,opinion), is designed to explicitly formulate this correspondence and kept in the prior covariance matrix. To do so, in the training corpus, extract *"subject and object"* pair, *"subject and predicate"* pair, *"modifier and subject"* pair from a positive sentence (opinion). In the testing corpus, if one such pair is observed, the corresponding feature value is checked, i.e., set to 1.

Another important feature is word synonyms. This is because that if only "This movie is good" is observed in the training corpus, and has a sentence says "The film is really good" in the testing corpus, a good opinion detection algorithm should be able to detect the second sentence as an opinion, however, without synonym information, it is not possible to be done. In the setting of Gaussian logistic regression, each entry in the prior covariance matrix can be represented as a linear interpolation of several indirect features, similar to "target-opinoin" pair described above, whether two words are within the same Wordnet[4] synset is also treated as a feature to reflect in the covariance matrix. More specificly, if two words appearing in the same Wordnet synset of the first sense of either noun, verb or adjective, their corresponding feature values is checked to 1.

By considering the two above word pair features, the feature vector F discussed and appeared in equation 13 can be written as:

$$F_{ij} = [1, CO_{ij}, S_{ij}, TO_{ij}] \tag{14}$$

where CO_{ij} is the log cooccurence of two word i and j within sentences. S_{ij} is 1 if two words i and j are in the same Wordnet synset), 0 otherwise. TO_{ij} is 1 if two words i and j are a target-opinion pair.

[2]We used the Sedumi QP solver (http://sedumi.mcmaster.ca) in the Yalmip (http://control.ee.ethz.ch/ joloef/yalmip.php) package.

5. DATASETS

TREC 2006 Blog corpus is used in this research. It contains 3,201,002 blog articles (TREC reports 3,215,171), is posted during the period of December 2005 to February 2006. The blog posts and the comments are from Technorati, Bloglines, Blogpulse and other web hosts.

Passage retrieval is performed to retrieve top 5,000 (or less than 5,000 if there is no more than 5,000 passages in the corpus for a particular query) passages for each of the 50 TREC Blog Opinion Retrieval topics. The search engine used in this research is Lemur, which retrieves 132,399 passages in total for 50 topics and 2,648 passages per topic in average. The retrieved passages are then separated into sentences and each sentence is classified as opinion or non-opinion sentence by Gaussian logistic regression with non-diagonal prior covariance as we reported in the preview sections.

There are two external datasets used in this research as training data. The first is a movie review dataset[3], prepared by Pang and Lee from Cornell University. There are 10,000 movie review sentences in this dataset in total, and 5,000 of them are positive examples, 5,000 are non-opinions. All the movie reviews are extracted from the Internet Movie Database(IMDb[4]).

The other external dataset is a product review dataset[5], prepared by Hu and Liu from University of Illinois at Chicago. There are more than 4,000 product review sentences, among them 2,034 are opinions, 2,173 are non-opinions. Those product reviews are extracted from customer comments about 2 brand digital cameras (Canon G3, Nikon coolpix 4300), 1 brand jukebox (Creative Labs Nomad Jukebox Zen Xtra 40GB), 1 brand cellphone (Nokia 6610) and 1 brand DVD player (Apex AD2600 Progressive-scan DVD player). As we can see here, they are mostly reviews about electronic products.

6. EXPERIMENTAL RESULTS

The main purpose of the experiments is to test whether the transfer learning approach used in this research is more effective on opinion detection than two other transfer learning methods. Given that we have no training data from the blog corpus, it is not possible to have a "real" baseline with training on the blog dataset and test on the same dataset. Therefore, the baseline system used in the experiments is a Gaussian logistic regression model trained on an external dataset and tested directly on the target dataset - blog dataset with zero mean, equal variance prior for regularization. This method is described in more details in section 4.

Another purpose is to explore the effectiveness of different settings for using the current approach. For example, we know that transfer learning is helpful in the case of no train data in a certain domain, but how to choose a good external dataset as the auxiliary domain? Do multiple external datasets improve the prediction accuracy more than a single one since based on what usually happens in non-transfer learning that the more the training data, the better the prediction performance. Another example, since we do not directly use word features in calculating the non-diagonal prior covariance, what will be the good indirect features for calculating it? Is Wordnet synset feature is better than target-opinion feature (see section 4)? The experiments conducted in this research will answer them in the following sections.

[3]http://www.cs.cornell.edu/People/pabo/movie-review-data/

[4]http://www.imdb.com/

[5]http://www.cs.uic.edu/~liub/FBS/FBS.html

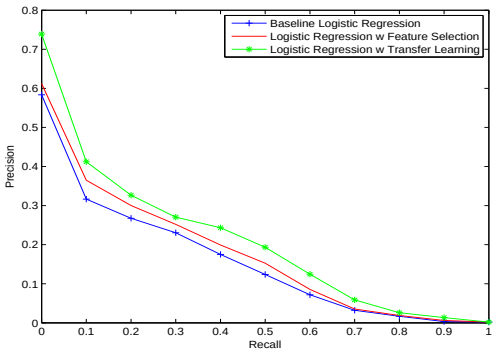

Figure 1: Comparison of Different Settings of Logistic Regression

Table 1: Mean Average Precision of Transfer Learning Approaches

Transfer Learning Approaches	MAP
Baseline	0.1657
Simple Feature Selection	0.1844
Our Approach	0.2190

The evaluation metric used in the experiment are precision at different recall level and mean average precision (MAP). The answers are provided by TREC qrel, which gives the document numbers of those documents containing an opinion and is related to the Blog opinion retrieval topics. Note that our system is developed for opinion detection at sentence level, and an averaged score of all the sentences in a retrieved passages, which is a part of a document, is returned as the final score. Therefore, to use TREC qrel to evaluate, we simply extract the unique document numbers that appearing in our returned passages, which is ranked by regression prediction score.

6.1 Effects of Using Non-diagonal Covariance Prior

This experiment compares the following three settings :

* Baseline: Using movie reviews to train the Gaussian logistic regression model with zero mean and equal variance. Vocabulary is unigram and bigrams from movie reviews. The model is directly tested on blog review data without any feature selection.

* Simple feature selection: Using movie reviews and product reviews to train the Gaussian logistic regression model with zero mean and equal variance. Vocabulary is the common unigram and bigrams from both domains. The model is test

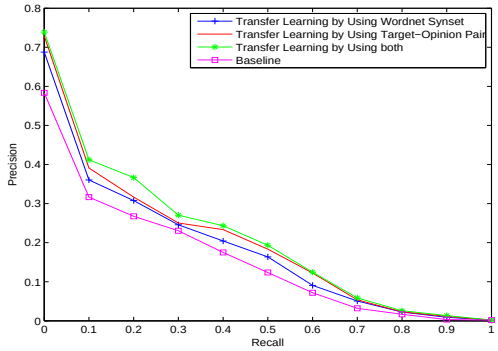

Figure 2: Impact of Different Features

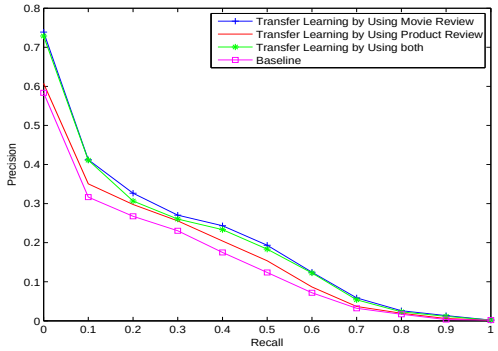

Figure 3: Impact of Different External Training Datasets

Table 2: Mean Average Precision of Transfer Learning Using Different Features

Features Used	MAP	Improvement
None(Baseline)	0.1657	-
Wordnet Synset Alone	0.1945	17%
Target-Opinion pair Alone	0.2114	28%
Both	0.2190	32%

on blog review data.

* The proposed approach: Using movie reviews to calculate prior covariance, train the logistic regression model with the informative prior. Vocabulary is from the blog corpus and is different for each retrieval topic based on the unigram and bigrams in the 5,000 retrieved passages. The model is test on blog review data.

Figure 1 shows the precision at each recall level for the tested three approaches. As we can see here, the approach used in this research gives the best precision at all the 11-point recall levels. The simple feature selection method also performs better than the baseline system, which indicates that by removing the bias introduced by a single domain of data, the prediction accuracy of transfer learning is improved. It is also obvious that the current approach is a more advanced way of learning task-related common knowledge than just doing simple feature selection.

Table 6.1 shows the non-interpolated mean average precision of the 3 approaches. Based on previous research [11] reported, the proposed approach could achieve an improvement of 20%-40% for text classification task. As for our task, we see an improvement of 32% on non-interpolated mean average precision from the baseline to the current approach. Both experiments in opinion detection and text classification show that construct non-diagonal prior covariance matrix to incorporate the external knowledge is a good way to boost the performance of gaussian logistic regression for transfer learning.

6.2 Effects of Feature Design

Target-opinion word pairs and Wordnet synonyms are two main features used in this project. It is reported that Wordnet synset feature is very effective for text classification task [11][10], by just using that, a 20%-40% improvement on text classification could be observed. Due to that opinion detection is using text classification techniques, so that it should be able to observe the similiar effects. However, opinion detection is not purely text classification, it is not topic-wised classification, but a binary classification of opinions or non-opinions. Therefore, Wordnet synset feature may not effective to our task. In section 4, we introduce a specific feature specially designed for the task of opinion detection, which is "Target-Opinion" word pairs. Each opinion is about a certain target, and this target usually has its own customary way to expression the opinion about it. There is a clear relationship between the target and the opinion about it. Is this a good feature as what we expected?

Figure 2 shows the results of an experiment which compares the three cases of using just Wordnet synset to create informative prior, using just target-opinion pairs to create informative prior and using both of them. It can be seen that applying the proposed approach with "Target-opinion" pair as the single feature is doing better than using Wordnet synset alone. When both features are used to construct the informative prior covariance, MAP reaches the best performance which the current approach in this research can achieve. Table 6.2 shows that using target-opinion pair alone, there is a 27% improvement as compared to the baseline and 10% more improvement as compared to using Wordnet synset alone. It proves that our hypothesis is correct. "Target-opinion" feature is more suitable for the task of opinion detection. Wordnet synset feature also contributes to the improvement of overall performance, but sometimes, for example at recall level 0.3 in Figure 2, there is no improvement from baseline to using Wordnet synset alone. It is not saying that this is a bad feature, but give us a hint that sometimes, Wordnet synset will not always be effective for the task of opinion detection.

6.3 Effects on External Dataset Selection

In our TREC-2006 submission, we selected common unigram and bi-gram features from both movie review and product review domains, with the belief that the intersection part could capture the common features across different domains as long as the task is the same, in this case, opinion detection. It is natural to extend this thought to apply

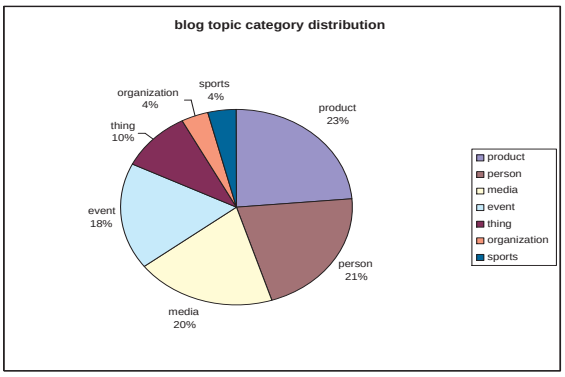

Figure 4: TREC-2006 Blog Topic Category

it into the approach used in this research, i.e., using both movie reviews and product reviews to train the Gaussian logistic regression model and also using both of them to generate prior.

Figure 3 shows the mean average precision at 11-point recall level for applying current approach with different external datasets. Surprisingly, using movie domain alone gives the best performance. Using product reviews to train the model results a performance drop as compared with using both domains, which not show an additive improvement as we expected. In this case, the negative effect of transfer learning is observed. It tells us that even transfer learning is effective, but sometimes it will not help much if a bad external training dataset is selected.

In our case, blog domain (target domain) covers more general topics as shown in Figure 4, movie domain (training domain) talking about mainly movies, but also talking about the people, objects, organizations in the movie, and hence matches blog domain better. On the other hand, product domain concentrates on customer reviews about several electronic products, it only helps a certain type of topics in blog opinion detection, not all of them. The experiment tells us that selecting a good external dataset is very important to avoid negative effect of transfer learning.

7. CONCLUSIONS

This paper describes a transfer learning approach which incorporates common knowledge for the same task from external domains as a non-diagonal informative prior covariance matrix. It brings a way to solve the problem of lacking of enough training data or even no training data from the target domain.

The approach is adapted to the task of opinion detection, which is a very interesting research topic recently. In our TREC-2006 system, opinion detection is separated into two sub-tasks, passage retrieval and text classification. Passage retrieval engine searches passages related to the query topics and return them by the confidence score. Text classification is a binary classification problem, either opinion or non-opinion. Sentences are the unit to perform this classification. Gaussian Logistic Regression is used as the general framework. In the proposed approach, an informative prior covariance matrix is constructed by incorporating external knowledge of "Target-Opinion" word pairs and Wordnet synset information. The results shown in the experiments prove that this is an effective approach with the fact that it achieves an 32% mean average precision improvement over baseline.

There are two main contributions of this work to the general communities of machine learning and opinion detection: first, solve the problem of with no labelled training data how to performing opinion detection for certain domains, second, study and extend transfer learning to opinion detection and explore important features for this task.

The future work will be a natural extension of the current work. In the experiment about the effect of different external datasets, we found that different datasets actually help the precision of opinion detection of different blog topics. Therefore, if we do blog topic classification and then use different external datasets as training data for each topic category, a greater improvement from the baseline should be observed.

8. REFERENCES

[1] L. L. B. Pang and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *proceedings of 2002 conference on Empirical Methods in Natural Language Processing*. EMNLP, 2002.

[2] J. Baxter. A bayesian/information theoretic model of learning to lear via multiple task sampling. In *Machine Learning*. Machine Learning, 1997.

[3] L. L. Bo Pang. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *proceedings of ACL 2004*. ACL, 2004.

[4] R. C. B. J. F. M. B. M. C. C. F. J. G. S. H. M. A. H. G. H. D. A. J. R. K. K. T. K. S. L. C. L. G. A. M. K. J. M. D. M. N. N. U. P. P. R. D. S.-O. R. T. R. P. v. d. R. E. V. Christiane Fellbaum, Reem Al-Halimi. *WordNet: An Electronic Lexical Database*. MIT Press, May 1998.

[5] J. W. E. Riloff. Learning extraction patterns for subjective expressions. In *proceedings of the 2003 conference on Empirical Methods in Natural Language Processing*. EMNLP, 2003.

[6] J. C. Hui Yang, Luo Si. Knowledge transfer and opinion detection in the trec2006 blog track. In *Notebook of Text REtrieval Conference 2006*. TREC, Nov 2006.

[7] C. M. G. M. I. S. Iadh Ounis, Maarten De Rijke. Overview of the trec-2006 blog track. In *Notebook of Text REtrieval Conference 2006*. TREC, Nov 2006.

[8] A. S. K. Yu, V. Tresp. Learning gaussian processes from multipl tasks. In *Proceedings of ICML 2005*. ICML, 2005.

[9] J. C. P. N. D. Lawrence. Learning to learn with the informative vector machine. In *Proceedings of ICML 2004*. ICML, 2004.

[10] T. Z. R. Ando. A framework for learning predictive structure from multiple tasks and unlabeled data. *ACM Jounal of Machine Learning Research*, May

2005.
[11] A. Y. N. Rajat Raina and D. Koller. Transfer learning by constructing informative priors. In *Proceedings of the Twenty-second International Conference on Machine Learning*. ICML, March 2006.
[12] K. T. T. F. S. Morinaga, K. Yamanishi. Mining product reputations on the web. In *Proceedings of SIGKDD 2002*. SIGKDD, 2002.
[13] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Proceedings of NIPS 1996*. NIPS, 1996.
[14] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL 2002*. ACL, July 2002.

APPENDIX

A. TREC-2006 BLOG TRACK TOPICS

march of the penguins
larry summers
state of the union speech
ann coulter
abramoff bush
macbook pro
jon stewart
super bowl ads
letting india into the club
arrested development
mardi gras
blackberry
netflix
colbert report
basque
whole foods
cheney hunting
joint strike fighter
muhammad cartoon
barry bonds
cindy sheehan
brokeback mountain
bruce bartlett
coretta scott king
american idol
life on mars
sonic
jihad
hybrid car
natalie portman
fox news report
seahawks
heineken
qualcomm
shimano
west wing
world trade organization
audi
scientology
olympics
intel
jim moran
zyrtec
board chess
oprah
global warming
ariel sharon
business intelligence resources
cholesterol
mcdonalds