Transfer Learning for Stack Exchange Tag Prediction

Abhishek Jindal [1] **Abhishek Walia** [1] **Siddharth Varia** [1]

Abstract

In this paper, we explore various transfer learning algorithms to tackle the task of predicting appropriate tags for questions posted on a website called Stack Exchange. For a given set of labeled data that contains titles, contents and tags for questions that belong to topics like biology, cooking, travel etc., the task is to predict tags for new questions from unseen domains, such as robotics. This task requires learning the relationship between different domains and using these relationships for prediction. To capture this relationship among different domains, we applied structural correspondence learning (SCL). In SCL, the important step is to learn good pivot features which are common to both the source and target domain and use these features to learn correspondence between features in the source and target domains. We also applied Latent Dirichlet Allocation (LDA) and CRF tagging to this task. We obtained the best F1 score of 17.68 by augmenting the predictions of CRF model with the predictions obtained from TF-IDF baseline.

1. Introduction

As sensors are becoming pervasive, more and more data is collected everyday, but such data is usually not labeled. Therefore, there arises a need to develop techniques in order to transfer knowledge from a different domain to label such data. Frequency feature bias is a very big challenge in transfer learning because of words that are domain specific are used more frequently in one domain because of the strong association with that topic domain. Negative transfer is another challenge while developing a transfer learning model, and occurs when the information learned from the source task in the source domain is detrimental to the target task in the target domain. Context feature bias presents yet another challenge as any word can have different meanings in different domains.

Table 1. Number of questions per topic

TOPIC	NUMBER OF QUESTIONS
BIOLOGY	13196
COOKING	15404
CRYPTOGRAPHY	10432
DIY	25918
ROBOTICS	2771
TRAVEL	19279

2. Related Work

Xue et al. extended the traditional probabilistic latent semantic analysis (PLSA) algorithm to integrate labeled and unlabeled data from different but related domains, into a unified probabilistic model. The new model was called Topic-bridged PLSA, or TPLSA. The limitation of this method that it assumes that prediction space is the same for both source and target domains.Therefore, PLSA is extremely useful for tasks like sentiment analysis where prediction space is positive, negative and neutral in both source and target domains. This assumption is not true for our task as the prediction space in our tasks in almost disjoint for source and target domains. Pan et al. proposed TCA to learn a low dimensional space to reduce the difference of distributions between different domains for transductive transfer learning in a computationally efficient manner.This method also suffers from the same prediction space assumption. It also suffers from a problem that it assumes that conditional distribution of words is similar across source and target domains. However, for domains like physics and cooking, this assumption is not valid.

Blitzer et al. proposed structural correspondence learning and successfully applied it to the task of cross-domain part of speech tagging.

*Equal contribution [1] Computer Science Department, Columbia University, New York, USA.

3. Data

The dataset we used for this problem was released as a part of a Kaggle competition. The data set contains question title, question content and associated tags for 6 different topics. These topics are biology, Robotics, DIY, Cryptography, Cooking & Travel. The testing part of the dataset comprise of just question title and question content for he Physics domain. The task is to predict the most appropriate tags associated with each question in the Physics domain based on the title and content of the question. Each question consists of a title, the HTML markup of the question body and the tags of the question.
In order to make the data useful for learning, we had to do considerable preprocessing. Since the Kaggle dataset was created by scraping StackExchange, the data had many HTML tags which we had to prune from the text. We cleaned out all punctuation except for hyphens and we removed the stop words which occurred too frequently like "the", "a", "an" which weren't entirely useful for prediction. We trimmed out any additional whitespace in the data. The data has a huge variance in the number of tags associated with a single question and it varies from 0 to 5.

Table 1 describes the number of questions per topic

4. TF-IDF Baseline

We started with tf-idf based approach directly on our target domain "Physics" as our baseline. TF-IDF scoring is one of the most basic weighting scheme in information retrieval and natural language processing. Below we have included the TF-IDF equations for the sake of completeness. To tag any question, we select top 4-6 words in either the question title, content or both to be its tags.

$$\begin{aligned} tf(t,d) &= 0.5 + 0.5 * \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}} \\ idf(t,D) &= \log \frac{N}{|\{d \in D : t \in d\}|} \\ tfidf(t,d,D) &= tf(t,d) * idf(t,D) \end{aligned} \tag{1}$$

5. Structural Correspondence Learning

Discriminative methods are very useful in machine learning when the train and test data come from the same underlying distribution. However in situations where the labeled data is available in one domain but labeled data is scare or unavailable for a target domain and the target domain data comes from a different distribution, one needs to apply transfer learning where the goal is to learn and induce correspondence among features from different domains. In our work, we apply Structural Correspondence learning (SCL) to learn such correspondence. In SCL, one uses unlabeled data from both domains to learn feature representations common to both the domains. Then a discriminative model is trained on these common features using labeled data from the source domain. The assumption is that such a discriminative model will generalize well enough to target domain because of the presence of the common features. These common features are referred to as pivot features and have a distributional and syntactic similarity in both the domains.

The first step in SCL is pivot feature selection. These pivot features are then used to learn a projection matrix θ which maps original features from both domains to the transformed features in the low dimensional feature space. These transformed features, hence created, are common to both the domains. Section 5.1 briefly describes the pivot selection process.

In a supervised setting, we utilize the original source domain features and transformed features in the source domain to learn a discriminative model which is used to predict tags in the target domain using target domain features and transformed target domain features.

Algorithm 1 Modified SCL for tag prediction

Input:
Labeled data from source domain, S = $\{(x_i, y_i)_{i=1}^{p}\}$
Unlabeled data from target domain, T = $\{(x_i)_{i=1}^{q}\}$
Output:
Classifier $f : X \to Y$
1. Split source domain data into two parts:

   1. Labeled source domain data, S_L = $\{(x_i, y_i)_{i=1}^{k}\}$
   2. Unlabeled source domain data, S_U = $\{(x_i)_{i=1}^{p-k}\}$

2. Choose m pivot features.
3. Create m binary classification problems using $S_U \cup T$
4. Get weight vector for each pivot feature:
for $l = 1$ **to** m **do**
$\hat{w_l} = argmin(\sum_j L(wx_i, p(x_i)) + \lambda||w||^2)$
end for
Where L is the loss function and $p(x_i)$ is 1 if pivot feature m occurs in x_i, otherwise it is 0.
5. Calculate θ:
$W = [\hat{w_1}|\hat{w_2}|\hat{w_3}|\cdots|\hat{w_m}]$
$[UDV^T]$ = SVD(W)
$\theta = U^T_{[1:h,:]}$
6. Return f, a classifier trained on S, $\{(\begin{bmatrix} x_t & \theta x_i \end{bmatrix}, y_i)_{i=1}^{p}\}$

The projection matrix θ is learnt by training classifiers to predict presence of pivot features. One classifier is learned for each pivot feature.To learn the pivot classifier, the pivot feature is removed from the training instances and then then

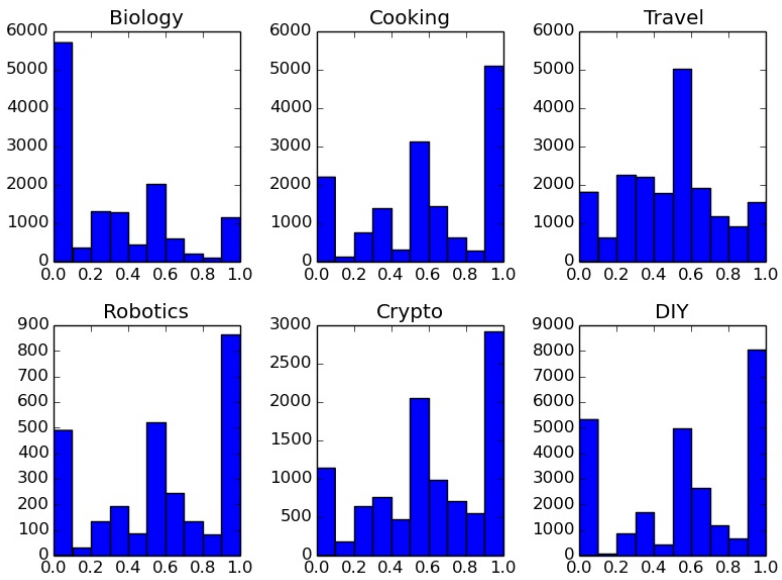

Figure 1. Binning of fraction of tags in question title and content. Y-axis represents number of questions and the X-axis represents equally sized bins between 0.0 and 1.0.

goal of this task is to predict whether pivot feature was present or absent in the training instance.

The weight vector of the learned pivot classifier encodes the covariance of the non-pivot features with the pivot features. Let's consider l_{th} pivot feature and k_{th} non-pivot feature of the classifier corresponding to this pivot feature. If the learned weight for this k_{th} feature is positive, then there is positive correlation between these features.

Given that the pivot features are common to both the domains, if the non-pivot features from both domain are positively correlated with many pivot features, then there is a high degree of correspondence between them.

By training these pivot classifiers, a projection matrix W is obtained, whose columns correspond to the learned weights of these pivot classifiers. For both computational and statistical reasons, a linear approximation of W is used by performing singular value decomposition of W, and then only the top h left singular vectors of W are used.
Let

$$W = UDV^T \tag{2}$$

then

$$\theta = U^T_{[1:h,:]} \tag{3}$$

Thus θ is the matrix whose rows are top left singular vectors of W. Let t be a training instance then by using original features x_i and projected features θx_i, a classifier is trained on the source domain with the assumption that it will generalize well in the target domain. If the mapping θ has captured enough correspondence between features in the source and target domain, the classifier trained on source domain will perform well in target domain.

5.1. Pivot Feature Selection

1: Pivot Words, P = {}
2: **for** words in each domain d **do**
3: Compute Mutual information (MI)
4: Sort words in descending order by MI and select top k words as domain specific words
5: $P = P \cup P_d$ where P_d = top k words in ascending order by MI
6: **end for**

6. Latent Dirichlet Allocation based unsupervised approach

LDA (Blei et al., 2003) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. LDA assumes that the documents and the words are derived from a generative process which is described below.

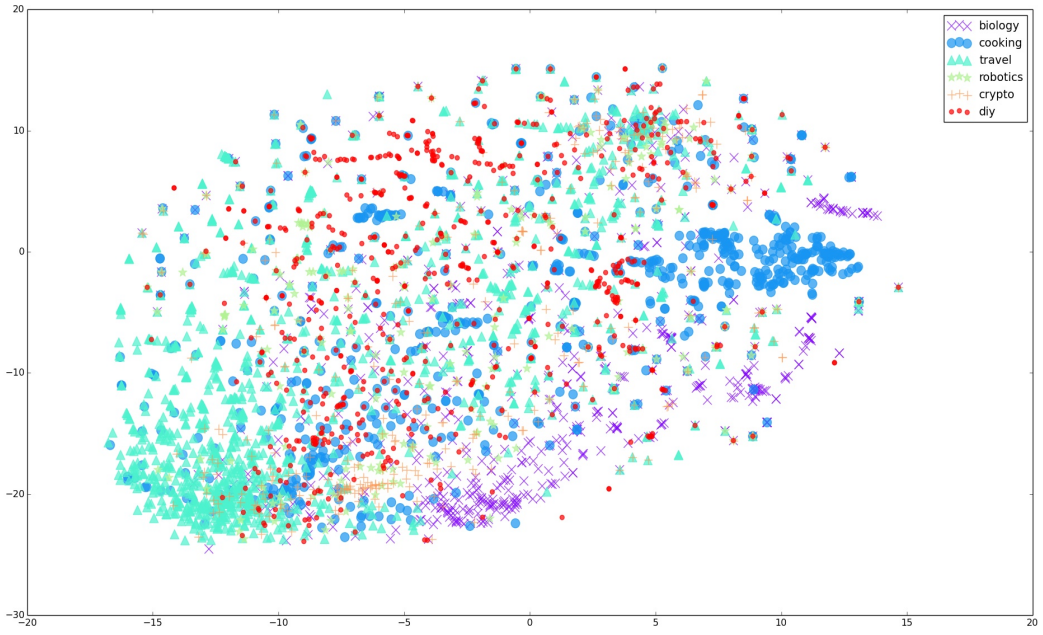

Figure 2. tSNE visualization of word embeddings of topics in different domains

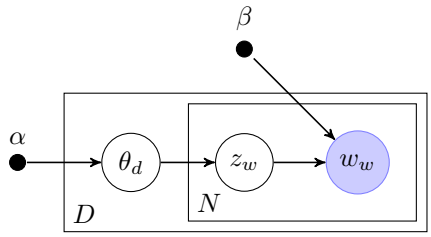

Figure 3. Diagram of LDA

6.1. Generative Process

1: **for** document d_d in corpus D **do**
2: Choose $\theta_d \sim$ Dirichlet(α)
3: **for** position w in d_d **do**
4: Choose a topic $z_w \sim$ Multinomial(θ_d)
5: Choose a word w_w from $p(w_w|z_w, \beta)$, a multinomial distribution over words conditioned on the topic and the prior β.
6: **end for**
7: **end for**

Our approach is to assume that each domain is composed of K topics and therefore, using LDA, we find the topic distribution for each domain, and for each topic we also have a distribution over words which denotes how associated is a particular word with that topic.

Algorithm 2 Unsupervised Tag Prediction using Latent Dirichlet Allocation

Input:
Unlabeled data from target domain, T = $\{(x_i)_{i=1}^q\}$
Output:
Set of tags $\{(y_i)_{i=1}^q\}$
Algorithm:

1. Extract K topics from the document using Gibbs-Sampling LDA:
2. For each question $\{(x_i)_{i=1}^q\}$
   (a) Get the topic distribution for the question.
   (b) For the topics having 2 highest probabilities
   (c) Select the top 5 words from each of these top 2 topics
   (d) Out of the above 10 words, predict those words as tags having maximum word similarity to the document.

As described in Algorithm 2, we extracted K (chosen as 10 for our experiments based on some hyperparameter experimentation) topics using LDA. Then for predicting the most

relevant tags, for each question, extracted the most prevalent topics for that question, and used the most informative words for the most prevalent topics. Out of the extracted words, we predicts those words which have maximum similarity with the words present in the question.
We observed that when we extracted the topics only the title of the question, the predictions were better compared to when we combined the title and the content of the question for extracting the topics. Studying some predictions showed us that using the content added some unwanted noise to the topic distribution and the word distributions, which led to a decline in our prediction performance.

7. Transfer learning using pos-tags and CRF (Conditional Random Fields)

Sentence 1: could(MD) someone(NN) recommend(VB) an(DT) introductory(JJ) book(NN) on(IN) **epidemiology**(NN)
Pos tags: MD NN VB DT JJ NN **IN NN Tag:** epidemiology

Sentence 2: introductory(NN) books(NNS) about(IN) **evolution**(NN)
Pos tags: NN NNS **IN NN**
Tag: evolution

Figure 4. For question titles ending with **IN NN**, the last word is usually a tag.

In order to learn to predict tags for the target domain physics using the structure of sentences in the source domain, say cooking, we use the algorithm described in Algorithm 3. Using the structure of sentences in the cooking domain, we train a CRF (Lafferty et al., 2001) classifier to predict whether a word is a tag candidate or not. A tag candidate is defined as a word that can possibly be an actual tag associated with a particular train instance. Then pointwise mutual information (PMI) of each candidate tag is used to select top n candidate tags with highest PMI as tags.

$$\text{PMI}(word, domain) = \log(\frac{P(word, domain)}{P(word) * P(domain)}) \tag{4}$$

Using this technique, we were able to predict tags for Physics domain, by using training data for cooking.

8. Recurrent Neural Network based approach

We also implemented multi-label classification using recurrent neural networks. Recurrent neural networks have revolutionized the field of NLP in recent years after the availability of word embeddings. Our model uses LSTM cells which address the vanishing gradient problem commonly found in RNNs. At each time step, an LSTM maintains a hidden vector $\mathbf{h}$ and a memory vector $\mathbf{c}$ responsible for controlling state updates and outputs. More concretely, we define the computation at time step t as follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f * \mathbf{h}_{t-1} + \mathbf{I}_f * \mathbf{x}_t + b_f) \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i * \mathbf{h}_{t-1} + \mathbf{I}_i * \mathbf{x}_t + b_i) \\
\widetilde{\mathbf{c_t}} &= \tanh(\mathbf{W}_c * \mathbf{h}_{t-1} + \mathbf{I}_c * \mathbf{x}_t + b_c) \\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \widetilde{\mathbf{c_t}} \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o * \mathbf{h}_{t-1} + \mathbf{I}_o * \mathbf{x}_t + b_o) \\
\mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{c}_t)
\end{aligned} \tag{5}$$

here σ is the logistic sigmoid function, $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_c, \mathbf{W}_o$ are recurrent weight matrices and $\mathbf{I}_f, \mathbf{I}_i, \mathbf{I}_c, \mathbf{I}_o$ are projection matrices. We customized the loss function because our task is to predict multiple tags instead of just one tag. We

Algorithm 3 Transfer learning using pos-tags and CRF

Input:
Labeled data from source domain, S = $\{(x_i, y_i)_{i=1}^p\}$
Unlabeled data from target domain, T = $\{(x_i)_{i=1}^q\}$
Train:
1. For each $(x_i, y_i) \in S$,

   1. For each word in x_i, create the following features to get x_i':

      (a) Previous 3 and next 3 part-of-speech tags along with part-of-speech tag of current word (context window of 7 words).

      (b) 2-gram, 3-gram and 4-gram combinations of part-of-speech tags in this context window.

   2. Convert y_i to a binary vector y_i', where $y_{ij} = 1$ iff word j in question i is a tag and 0 otherwise.

2. Train a CRF classifier f using $\{(x_i', y_i')_{i=1}^p\}$
Predict:
1. For each word in $x_i \in T$, create the following features to get x_i':

   1. Previous 3 and next 3 part-of-speech tags along with part-of-speech tag of current word (context window of 7 words).

   2. 2-gram, 3-gram and 4-gram combinations of part-of-speech tags in this context window.

2. Predict binary vector $y_i' = f(x_i')$
3. Use the predicted binary vector y_i' to find out tag candidates in x_i

Table 2. F-1 measure using various methods

METHOD	TITLE ONLY	CONTENT ONLY	TITLE + CONTENT
TF-IDF BASELINE	8.30	6.38	7.32
UNSUPERVISED TAGGING USING LDA	6.70	4.90	5.50
UNSUPERVISED TAGGING USING POS TAGS AND CRF	11.50	7.28	8.45
SCL	10.03	6.52	8.74
UNSUPERVISED TAGGING USING POS TAGS AND CRF + TFIDF	17.68	11.27	13.12

take softmax over all the tags in the domain and pick the top k tags based on their posterior probabilities. Because titles worked best with TF-IDF and LDA based approaches, we used only titles in our experimentation with RNNs.

9. Results and Conclusion

In Table 2, we report the F1 score we obtained with different methods described in the paper. As we can see LDA did not do well and this can be attributed to the fact that for many of the questions, the tags do not appear in either the title or the content of the question. We obtained the best F1 score of 17.68 when we combined the tags discovered by CRF along with those discovered by TF-IDF weighing scheme. Even though we implemented recurrent neural network based multi-label classification system, due to lack of time, we were not able to carry out transfer learning using RNNs. We obtained accuracy of around 20% when we applied RNNs within the same domain by splitting the labeled data in to train and validation sets. Thorough experimentation using RNNs will be worth trying in the future.

References

Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent dirichlet allocation. volume 3, pp. 993–1022. JMLR.org, March 2003. URL http://dl.acm.org/citation.cfm?id=944919.944937.

Blitzer, John, McDonald, Ryan, and Pereira, Fernando. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pp. 120–128, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-73-6. URL http://dl.acm.org/citation.cfm?id=1610075.1610094.

Lafferty, John D., McCallum, Andrew, and Pereira, Fernando C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655813.

Pan, Sinno Jialin, Tsang, Ivor W., Kwok, James T., and Yang, Qiang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pp. 1187–1192, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1661445.1661635.

Xue, Gui-Rong, Dai, Wenyuan, Yang, Qiang, and Yu, Yong. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 627–634, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390441. URL http://doi.acm.org/10.1145/1390334.1390441.