

# Domain Adaptation for Sentiment Analysis using Keywords in the Target Domain as the Learning Weight

**Jing Bai      Hiroyuki Shinnou      Kanako Komiya**

Ibaraki University, Department of Computer and Information Sciences  
4-12-1 Nakanarusawa, Hitachi, Ibaraki JAPAN 316-8511

{17nm719x, hiroyuki.shinnou.0828, kanako.komiya.nlp}  
@vc.ibaraki.ac.jp

## Abstract

This paper proposes a new method of instance-based domain adaptation for sentiment analysis. First, our method defines the likelihood of keywords, through the value of inverse document frequency (IDF), for each word in documents in the target domain. Next, the keyword content rate of a document is calculated using the likelihood of keywords and the domain adaptation is performed by giving the keyword content rate to each document in the source domain as the weight. The experiment used an Amazon dataset to demonstrate the effectiveness of our proposed method. Although the instance-based method has not shown great efficiency, the advantages combining instance-based method and feature-based method are shown in this paper.

## 1 Introduction

This paper proposes a new method of instance-based domain adaptation for sentiment analysis. Sentiment analysis involves judging a polarity, positive or negative, of a review such as a movie review. This is one of the document classification tasks and supervised learning can be used to solve it. However, if the domain of the test data is different from the domain for the learning data (for example, book reviews), the accuracy of the classifier obtained through standard supervised learning reduced. This is the problem with domain shift.

The solution to this problem is domain adaptation. Domain adaptation can be roughly divided into two categories: feature-based and instance-based (Pan

and Yang, 2010). In summary, both are weighted-learning methods, but feature-based gives weights to features and instance-based gives weight to instance.

Here, we present a new instance-based method. Generally, the instance-based method assumes a covariate shift, and gives the weight based on the probability density ratio between target domain and source domain. However, the computational cost for the instance-based method is too high. The method presented here is simple and its effect is better than methods using a typical probability density ratio.

Our method first defines  $l_w$ , the likelihood of the keyword of the word  $w$  using the IDF in the target domain. Using  $l_w$ , the weight of a review  $x$  in the source domain is set as the keyword content rate  $w_x$ . After that, weighted-learning is performed by giving  $w_x$  to each document in the source domain  $x$  to overcome the domain shift.

In the experiment, we used Amazon dataset (Blitzer et al., 2007), and compared our proposed method with two typical instance-based methods: unconstrained least squares importance fitting (uLSIF) (Yamada et al., 2011) using the probability density ratio and the method defining weight through Naive Bayes model (Shinnou and Sasaki, 2014), to demonstrate the effectiveness of the proposed method.

## 2 Related Work

Domain adaptation is roughly divided into two types: the supervised approach using labeled data in the target domain and the unsupervised approach that does not use them. For supervised approach, Daumé's method (Daumé III, 2007) has become a

standard method because of its simplicity and high ability.

The method in the current research is an unsupervised approach. Unsupervised approaches can further be divided into two types: feature-based and instance-based (Pan and Yang, 2010). They are both weighted learning methods; feature-based methods give weights to features and instance-based methods give weights to instances. Among feature-based methods, the most representative method is structural correspondence learning (SCL) (Blitzer et al., 2006). In addition, CORAL (Sun et al., 2016) has attracted much attention for its simplicity and high ability in recent years. Moreover, the feature-based methods with deep learning (Glorot et al., 2011), the expanded CORAL (Sun and Saenko, 2016) and adversarial networks (Ganin and Lempitsky, 2015)(Tzeng et al., 2017) are also considered as the state of the art.

On the other hand, instance-based methods have not been studied as much as feature-based methods. The instance-based method assumes a covariate shift. A covariate shift assumes  $P_S(c|\mathbf{x}) = P_T(c|\mathbf{x})$  and  $P_S(\mathbf{x}) = P_T(\mathbf{x})$ . Under a covariate shift,  $P_T(c|\mathbf{x})$  can be obtained by the weighted learning that uses the probability density ratio  $r = P_T(\mathbf{x})/P_S(\mathbf{x})$  as the weight of the document of the source data  $x$ . There are a variety of methods for calculating the probability density ratio. The simplest way to calculate the ratio is directly estimate  $P_S(\mathbf{x})$  and  $P_T(\mathbf{x})$ , but in the case of complex models, the problem will be more complicated. Thus, the method that directly models the probability density ratio was studied. Among these methods, uLSIF (Yamada et al., 2011) is widely used because the time complexity of the method is relatively small. However,  $P(\mathbf{x})$  of bag-of-words can be modeled by Naive Bayes model if the problem is limited to natural language processing. Therefore, (Shinnou and Sasaki, 2014) defined  $P_r(\mathbf{x})$ , the prior of  $\mathbf{x}$ , as follows:  $P_R(\mathbf{x}) = \prod_{i=1}^n P_R(f_i)$ , where  $\mathbf{x}$  denotes a data in the domain  $R$  and  $\mathbf{x}$  has a set of features, that is,  $\mathbf{x} = \{f_1, f_2, \dots, f_n\}$ . They also obtain  $P_R(f_i)$  using the following equation:  $P_R(f) = \frac{n(R,f)+1}{N(R)+2}$ . Here,  $n(R; f)$  is the frequency of feature  $f$  in the domain  $R$ , and  $n(R)$  is the number of data in the domain  $R$ . Therefore, the probability density ratio

is obtained as follows:

$$r = \frac{P_T(\mathbf{x})}{P_S(\mathbf{x})} = \frac{n(T, f) + 1}{N(T) + 2} \cdot \frac{N(S) + 2}{n(S, f) + 1} \quad (1)$$

### 3 Proposed Method

#### 3.1 Likelihood of the Keyword in the Target Domain

The likelihood of the keyword in the target domain is  $l_x$ , and  $l_x$  is set as the value of IDF in the target domain of  $w$ :

$$l_x = \log \left( \frac{N}{d_i} \right) + 1$$

Here  $N$  is the number of articles in the article collection in the target domain, and  $d_i$  is the number of articles containing the word  $w$  in the article collection in the target domain.

#### 3.2 The Content of Keywords in the Source Case

Set the weight  $w_x$  of the instance  $x$  in the source domain. The words (file)  $x$  is  $\{w_i\}_{i=1}^K$ , and the frequency within  $x$  for word  $w_i$  is  $f_i$ . Using these,  $w_x$  is given by the following equation:

$$w_x = \frac{1}{\sum_{i=1}^k f_i} \sum_{i=1}^K f_i \cdot l_{w_i}$$

### 4 Experiment

The Amazon dataset (Blitzer et al., 2007) used in the experiment is specifically developed using the `processed_acl.tar.gz` file on the following website.

<https://www.cs.jhu.edu/~mdredze/datasets/sentiment/>.

The data include books (B), dvd (D), electronics (E), and kitchen (K). The number of files contained in each domain is shown in Table 2. There are 1000 positive data and negative data in each domain, and these 2000 data are used as training data in this domain.

Table 1: Experimental result

	IDEAL	NONE	uLSIF	NB	our method
B → D	0.822	0.806	0.806	<b>0.811</b>	0.809
B → E	0.852	0.761	0.756	0.755	<b>0.765</b>
B → K	0.878	0.845	0.778	0.779	<b>0.785</b>
D → B	0.831	0.762	0.733	<b>0.745</b>	0.741
D → E	0.852	0.761	0.748	0.753	<b>0.758</b>
D → K	0.878	0.795	0.773	0.782	<b>0.789</b>
E → B	0.831	0.712	0.714	<b>0.723</b>	0.719
E → D	0.822	0.722	0.708	<b>0.723</b>	0.714
E → K	0.878	0.849	0.854	<b>0.857</b>	0.855
K → B	0.831	0.713	0.707	0.714	<b>0.715</b>
K → D	0.822	0.740	0.733	0.723	<b>0.736</b>
K → E	0.852	0.842	0.847	<b>0.852</b>	0.845
Average	0.846	0.776	0.763	0.768	<b>0.769</b>

Table 2: The number of files in each domain

	positive	negative	test data
books	1,000	1,000	4,465
dvd	1,000	1,000	3,586
electronics	1,000	1,000	5,681
kitchen	1,000	1,000	5,945

The learning algorithm is an SVM with scikit-learn. The core is linear, the value of the  $c$  parameter is fixed at 0.1, and the scikit-learn SVM supports Weighted-Learning<sup>1</sup>, so the scikit-learn SVM is used here.

domain adaptations are:  $B \rightarrow D, B \rightarrow E, B \rightarrow K, D \rightarrow B, D \rightarrow E, D \rightarrow K, E \rightarrow B, E \rightarrow D, E \rightarrow K, K \rightarrow B, K \rightarrow D, K \rightarrow E$ . See Table 1 for the results of the two methods (uLSIF (Yamada et al., 2011) and using equation (1) of Naive Bayes) for determining the rate density ratio of each domain and the proposed method. “NONE” in Table 1 means that the domain adaptation method was not used but simply applies the classifier formed from the training data of the source domain to the result of the test data in the target domain was applied. In addition, IDEAL is a result using the training data in the target domain to learn through the classifier and apply it to the test data in the target domain.

Using the case as a weighted method, a compari-

<sup>1</sup>[http://scikit-learn.org/stable/auto\\_examples/svm/plot\\_weighted\\_samples.html](http://scikit-learn.org/stable/auto_examples/svm/plot_weighted_samples.html)

son of uLSIF, NB, the our method shows that the six highest correct answer rates in the 12 domain adaptations are obtained by our method, and the remaining six highest positive answer rates are obtained by NB. When we take 12 averages, the solution rate of our method is more than that of NB, and our method is weighted with example and, which is excellent.

## 5 Discussion

NONE in Table 1 is compared with the case weighting method (uLSIF, NB, and our method). It is clear that NONE has a high positive solution rate. For these data only, the instance-based method has no effect on domain adaptation.

However, feature-based and instance-based methods are easy to combine. Here, the four domains applied in paper (Sun et al., 2016) are adapted to  $B \rightarrow E, D \rightarrow B, E \rightarrow K$  and  $K \rightarrow D$ , and SCL conversion training is used first. The prime vector of the data, then experiment with the weighted-learning in this transformed vector is performed using the proposed method. The results are shown in Table 3. The CORAL in Table 3 is taken from (Sun et al., 2016).

From Table 3, it can be seen that SCL has no effect. After SCL is combined with our method, the accuracy is not high enough. However, when SCL is combined with the proposed method, the precision for SCL alone is improved. The positive effect of

Table 3: Combination of feature-based method and instance-based method

	IDEAL	NONE	CORAL	our method	SCL	SCL + our method
B → E	0.852	0.761	0.763	0.760	0.757	0.756
D → B	0.831	0.762	0.783	0.756	0.732	0.733
E → K	0.878	0.849	0.836	0.849	0.852	0.853
K → D	0.822	0.740	0.739	0.743	0.732	0.733
Average	0.846	0.778	0.780	0.777	0.768	0.769

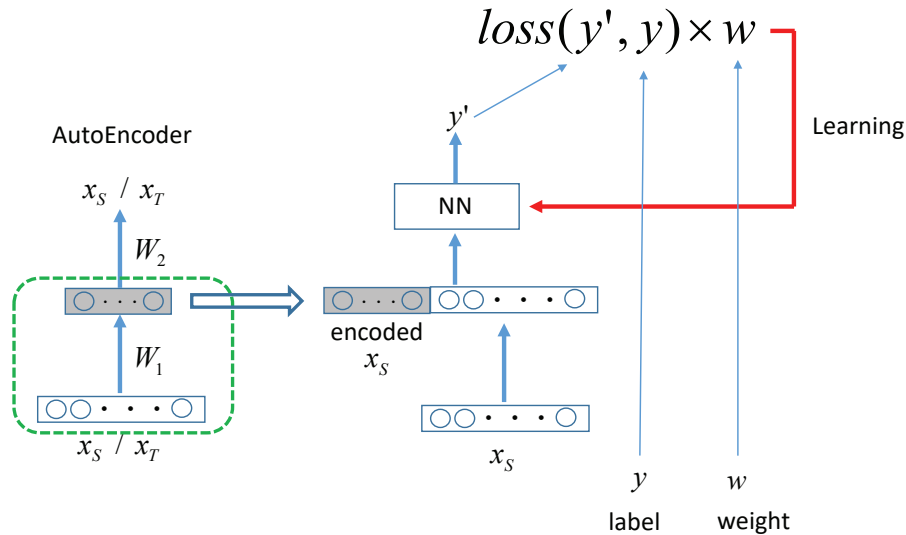


Figure 1: AE+NN+Weighted-Learning

the combined the instance-based and feature-based methods can be confirmed. There are many ways in using the feature-based method, in addition to SCL, so it can be improved by combining these techniques with the proposed method.

In addition, although the weighted-learning SVM is used in this paper, the loss value of the loss function in the neural network is multiplied by the weight, and it is easier to realize weighted-learning as the loss value. There are many options for using the domain adaptation method on deep learning, and these solutions combined with instance-based are easier to compute.

In a simple example, we used the AutoEncoder (AE) as the feature-based method. Using AE, the dimension of the data in the source and target domain was reduced, that is, encoded. In learning and testing, we used the connected data of the orig-

inal data and the encoded data instead of the original data. In learning, as described above, the value of the loss function was multiplied by the weight obtained by our method and was taken as the loss value (FIG. 1). Only the experiment of B → E is performed, and the results in Table 4 と FIG. 2 were obtained. In addition, in this experiment, neural network learning was ended in 50 epochs, and the correct rate was the result from evaluating the model obtained by learning data after 50 epochs. Moreover, the dimension was reduced to 200. Every method used the same multi-layer perceptron, which has three layers.

The use of the connected data of the original data and the encoded data is a feature-based method. FIG. 2 shows that this feature-based method NN+AE improves the precision of the standard neural network NN. Moreover, combining

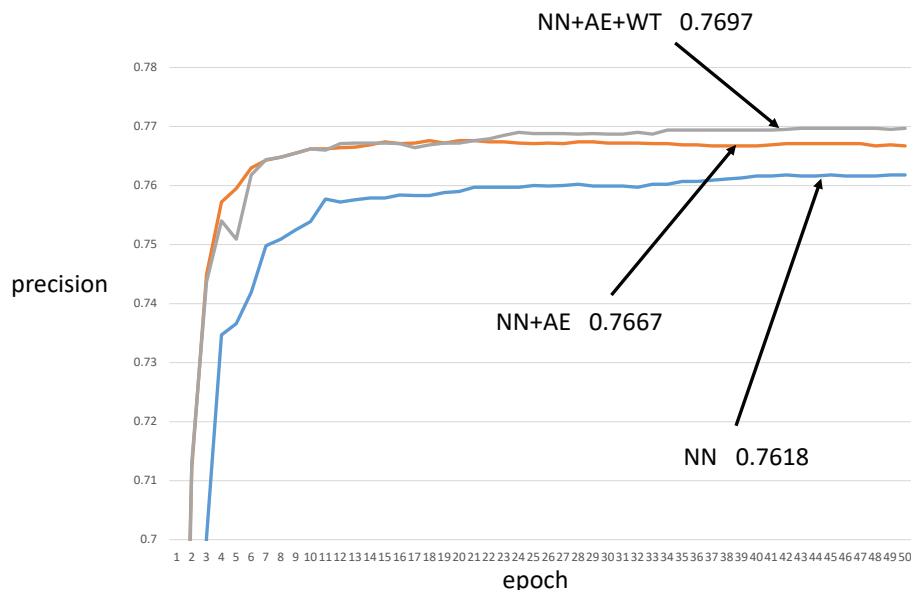


Figure 2: Weighted-Learning by neural network

Table 4: weighted-learning by neural network

IDEAL	NONE	NN	NN+AE	NN+AE+WT
0.852	0.761	0.7618	0.7667	0.7697

NN+AE and our method further improves it. This result shows that the combination of the feature-based method and the instance-based method is easy in network learning and effective. In the future, we are planning to design a domain adaptation method in this framework.

## 6 Conclusion

This paper proposed a method for instance-based domain adaptation of sentiment analysis. For outline, from the target domain, using IDF to set the likelihood of keywords, and the data in the source domain, the content rate in the target domain keyword, and the keyword content rate as the weight. In the experiment, we compared our proposed method with two typical instance-based methods: uLSI using the probability density ratio and the method defining weight through Naive Bayes model. However, using an instance-based alone to perform domain adaptation has a very small effect, the combining instance-based method and feature-

based method is assured as shown in this paper. Further, the combination is easy to implement in the neural network model. Thus, we will investigate this approach in future.

## References

- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP-2006*, pages 120–128.
- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain adaptation for Sentiment Classification. In *ACL-2007*, pages 440–447.
- Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *ACL-2007*, pages 256–263.
- Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *ICML-11*, pages 513–520.

- Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359.
- Hiroiyuki Shinnou and Minoru Sasaki. 2014. Domain Adaptations for Word Sense Disambiguation under the Problem of Covariate Shift (in Japanese). *Journal of Natural Language Processing*, 21(1):61–79.
- Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops*, pages 443–450.
- Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of Frustratingly Easy Domain Adaptation. *AAAI*.
- Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*.
- Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. 2011. Relative density-ratio estimation for robust distribution comparison. *Neural Computation*, 25(5):1370–1370.