Case Study of Model Adaptation: Transfer Learning and Online Learning

Kenji Imamura
NTT Media Intelligence Laboratories
1-1 Hikari-no-oka, Yokosuka, 239-0847 Japan
imamura.kenji@lab.ntt.co.jp

Abstract

Many NLP tools are released as programs that include statistical models. Unfortunately, the models do not always match the documents that the tool user is interested in, which forces the user to update the models.

In this paper, we investigate model adaptation under the condition that users cannot access the data used in creating the original model. Transfer learning and online learning are investigated as adaptation strategies. We test them on the category classification of Japanese newspaper articles. Experiments show that both transfer and online learning can appropriately adapt the original model if the dataset for adaptation contains all data, not just the data that cannot be well handled by the original model. In contrast, we confirmed that the adaptation fails if the dataset contains only erroneous data as indicated by the original model.

1 Introduction

Recent natural language processing (NLP) systems are built using machine learning (supervised learning). The developers of these systems basically create annotated corpora from which statistical models are generated. However, if the documents that users want to apply the systems to do not belong to the domain of the annotated corpora, the resulting accuracy tends to be unsatisfactory.

For instance, Figure 1 shows the typical drop in accuracy in the category classification task of newspaper articles over time; the statistical model was trained using supervised data from 1995 (details are described later). Even though the test data were obtained from newspaper articles (i.e., the same domain data), the accuracy against 2007 articles fell by about 10% from the 1996 articles. The reasons for this include the emergence of new words and changes in word distribution. In order to recover this degradation, we have to re-train the models.

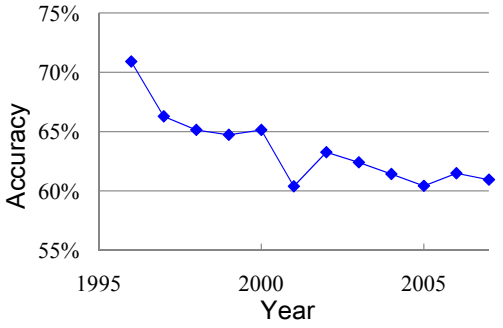


Figure 1: Accuracy of Category Classification with training by 1995 Dataset

To overcome this problem, transfer learning methods have been proposed (Pan and Yang, 2010). Many transfer learning methods assume that the users can obtain both the original data and additional data for adaptation. However, in most practical cases, the users sometimes are unable to access the original data. For example, only the developers are licensed to handle the original data, not the users.

NLP tools, such as taggers, parsers, and classifiers, are commonly released as programs that include the original models. Since many users cannot update the original models, they continue to use them even if the user's documents do not match the models (Figure 2).

The objective of this paper is to investigate methods that, given an additional dataset, permit adaptation of original models under the constraint that the original dataset is unavailable.

The target task of this paper is category classification of newspaper articles. Because NLP tools such as taggers or parsers are founded on struc-

International Joint Conference on Natural Language Processing, pages 1292–1298,
Nagoya, Japan, 14-18 October 2013.

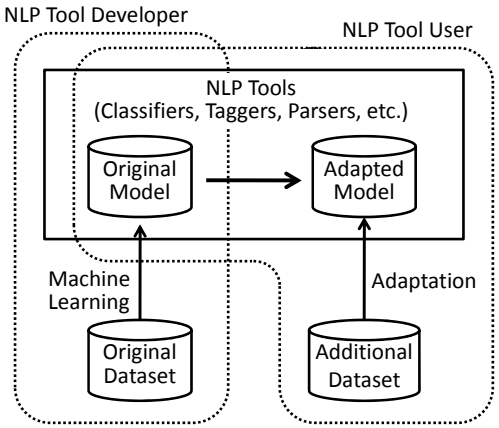


Figure 2: Relationship between Developers and Users of NLP Tools

tured learning, which extends the classification, we select the linear classification task.

In this paper, we investigate the combination of the following learning methods and additional datasets.

- We test two learning methods, batch and online learning. In batch learning, we use a maximum entropy classifier (Berger et al., 1996; Chen and Rosenfeld, 2000) and adapt the model using transfer learning. In online learning, we select soft confidence-weighted learning (Wang et al., 2012).

- We test two kinds of additional datasets. One is that all data are used for adaptation. The other is that only the data that failed to predict correct categories by the original model are used. We consider the active learning strategy for the second dataset.

The remainder of this paper is organized as follows. In Section 2, we detail the task, datasets, and learning methods (batch and online learning). Section 3 describes the experiments conducted and their results, and Section 4 summarizes the findings of this study.

2 Settings and Adaptation Methods

2.1 Task and Data

The task of this study is category classification of Japanese newspaper articles. We selected articles from Mainichi Shinbun newspapers for the years of 1995, 1996, 2005, and 2006. A part of the 1995 data is widely used in the Japanese NLP community because its dependency structures and predicate-argument structures have been annotated [1].

Set Name	Period	# of Data
Original Dataset	Jan.,1995 - Nov.,1995	102,454
Additional Dataset	Jan.,2005 - Nov.,2005	88,202
Development set	Dec.,1995	9,043
Test set 1	Jan.,1996 - Dec.,1996	114,116
Test set 2	Jan.,2006 - Dec.,2006	95,761

Table 1: Statistics of Data Used

One of 16 categories is assigned to each article. The category denotes type of the article, such as 'Economics', 'International', 'Sports', 'Top page', and so on. The task of this study is to predict the category of each article from its content (text).

Figure 3 shows the relationships among datasets (for learning and testing) and models. We took articles from Jan. to Nov. in 1995 as the original dataset, and used them to train the original model. The original dataset was not used thereafter. Articles from Dec. 1995 were used to tune the model's hyperparameters. The additional dataset for adaptation was created from articles from Jan. to Nov. 2005. We prepared two test sets. The first consisted of 1996 articles (Test set 1), and the second consisted of 2006 articles (Test set 2). Our objective is to improve the accuracy against Test set 2. The statistics of the datasets are shown in Table 1.

Features for classification are 'bag-of-words' of the title and the first paragraph of the article. Only content words (nouns, verbs, adverbs, adjectives, and interjections) that appear more than once are used as features.

2.2 Transfer Learning from Batch Learning

2.2.1 Regularized Adaptation

The problem setting of this paper is a sort of transfer learning (domain adaptation). Because we cannot access the original data, this problem is regarded as "model-based domain adaptation" according to the taxonomy of transfer learning by Sha and Kingsbury (2012). Regularized adaptation (Evgeniou and Pontil, 2004; Xiao and Bilmes, 2006) is a variant of model-based domain adaptation. As the regularizer, it uses the differences

[1] Dependency structures are published as Kyoto University Text Corpus (http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?Kyoto University Text Corpus). Predicate-argument structures are published as NAIST Text Corpus (Iida et al., 2007) (http://cl.naist.jp/nldata/corpus/). Note that the texts of the articles must be purchased from the newspaper company.

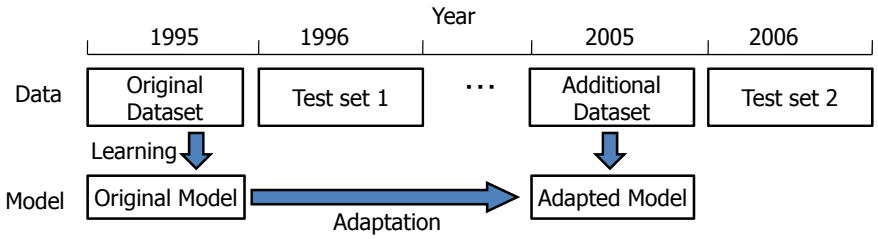


Figure 3: Datasets and Models

in parameters between the adapted model and the original model, not adapted parameters. This is done to minimize the differences between the original model and the adapted model.

Although Evgeniou and Pontil (2004) proposed regularized adaptation for SVMs, and Xiao and Bilmes (2006) proposed the same for neural networks, they can also be applied to maximum entropy classifiers [2]. The loss function, ℓ, is represented as follows.

$$\ell = -\sum_i \log P(y_{AD_i}|\boldsymbol{x}_{AD_i};\boldsymbol{w}_{AD}) + \frac{1}{2C}\sum_{k=1}^{d}(w_{AD_k} - w_{OR_k})^2, \quad (1)$$

where $P(y|\boldsymbol{x};\boldsymbol{w})$ denotes the posterior probability of a sample computed with the weight parameters of the model $\boldsymbol{w}$; y_{AD_i} and $\boldsymbol{x}_{AD_i}$ are the input and output of the ith sample in the additional dataset, respectively, w_{AD_k} and w_{OR_k} denote weight parameters of the adapted and the original model, respectively; both have dimensions of d, and C is a hyperparameter.

The maximum entropy classifier used in this paper estimates the weight parameters to minimize the above loss function. The first term in Equation (1) suppresses discriminative errors of the additional data at minimum, and the second term suppresses differences between the original model and the adapted model.

2.2.2 Regularization with Two Hyperparameters

The output classes of the adapted model are identical to those of the original model in this task. In contrast, features for classification are not identical because new words appear over time.

[2]Regularized adaptation is used as a re-training function of the Japanese morphological analyzer MeCab (Kudo et al., 2004), which is based on conditional random fields (CRFs). http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

In Equation (1), all features, which include features from the original data and the additional data, are treated equally. However, if we significantly change weight parameters of the original model features, the original model can correctly class less data due to errors. In contrast, with regard to the new features from the additional data, we can change the parameters without limitation. Therefore, it is natural to distinguish new features from those of the original model.

Here, assuming that the number of dimensions of the parameters in the original model is d_{OR}, and that in the adapted model (i.e., the features include the original and additional data) is d_{AD}, the loss function becomes,

$$\ell = -\sum_i \log P(y_{AD_i}|\boldsymbol{x}_{AD_i};\boldsymbol{w}_{AD}) + \frac{1}{2C_{AD}}\sum_{k=1}^{d_{OR}}(w_{AD_k} - w_{OR_k})^2 + \frac{1}{2C_{OR}}\sum_{k=d_{OR}+1}^{d_{AD}} w_{AD_k}^2, \quad (2)$$

where C_{OR} denotes the hyperparameter that was used while learning the original model, and C_{AD} denotes the hyperparameter for the additional data. If we set them as $C_{OR} \geq C_{AD}$, only the new features from the additional data can change significantly; changes to the existing features of the original model are suppressed.

2.3 Online Learning

Online learning is a strategy that updates current parameters in order to correctly classify training samples one-by-one. It matches the problem setting in this paper because it can train a new model by altering the original model to suit the additional data. However, it usually loses information about old samples (in our case, the original data). Therefore, we need to iterate the learning process on the entire dataset several times.

The recent proposal Confidence-weighted learning (CW) generates each weight parameter from a Gaussian distribution whose mean is μ and standard deviation is σ (Dredze et al., 2008; Crammer et al., 2009a). This method expresses confidence in frequently updated parameters, and accepts only small changes to them. Rarely updated parameters can be greatly changed. Confidence is expressed by a covariance matrix. CW is known to offer faster convergence than the conventional online learning algorithms such as perceptrons and passive-aggressive methods. In other words, CW makes learned samples hard to forget. The CW algorithm offers the possibility of adapting to the additional data without referring to the original data.

It is known that the training performance of the original CW algorithm suffers if the training samples contain significant noise components that are linearly-inseparable. The adaptive regularization of weight algorithm (AROW; (Crammer et al., 2009b)) and the soft confidence-weighted learning algorithm (SCW; (Wang et al., 2012)) were proposed to overcome this weakness. In this paper, we employ the SCW algorithm.

In SCW-I, which uses a linear penalty, parameter updating is represented as follows.

$$(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) = \arg\min_{\boldsymbol{\mu},\Sigma} \{D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \| \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)) + C\ell^{\phi}(\mathcal{N}(\boldsymbol{\mu}, \Sigma), (\boldsymbol{x}_t, y_t))\}, \quad (3)$$

where $\boldsymbol{\mu}$ denotes a mean vector and Σ denotes a covariance matrix of the parameters, $D_{KL}(\cdot\|\cdot)$ denotes Kullback-Leibler divergence, $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a multivariate normal distribution with mean of μ_k and standard deviation of σ_k, $\ell^{\phi}(\cdot)$ is a loss function based on the hinge loss, and C is a hyperparameter that restricts the maximum change permitted in the update. Following Equation (3), the loss of the correct class y_t predicted from input feature vector $\boldsymbol{x}_t$ becomes minimum by the second term, and simultaneously the change in parameters is suppressed by the first term. (Final update formulae are provided in (Wang et al., 2012)).

However, there are some problems in implementing Equation (3) directly. The following approximations are applied in general.

- Weight parameters $\boldsymbol{w}$ should be generated from Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, but the mean vector $\boldsymbol{\mu}$ is directly used as weight parameters.
- The size of the covariance matrix is $d \times d$, where d denotes the number of dimensions of the parameters, and so memory consumption is high. To avoid this problem, only diagonal elements are considered (the matrix is degenerated to a vector).

In addition, the hyperparameters that control the maximum change and the confidence value, C and ϕ, must be set manually.

To apply the SCW algorithm, we first construct the original model from the original data using Equation (3) until classification errors become minimum on the development set. Note that the original model retains not only the mean vector but also the covariance matrix. In adaptation, we regard the original model as $(\boldsymbol{\mu}_0, \Sigma_0)$ and similarly update it using the additional data, one-by-one.

3 Experiments

3.1 Experimental Settings

Methods We test the methods described in Sections 2.2 and 2.3 (represented as 'Transfer' and 'Online,' respectively). The following baselines are also tested.

(a) Original Model. This case yields the upper bound of Test set 1.

(b) The model is trained using only the additional dataset. If there is enough data, this yields the upper bound of Test set 2.

(c) The model is trained by the feature augmentation method (Daumé, 2007) using the original data and the additional data, which is one of the domain adaptation techniques. This case yields the upper bound if we can access the original data.

(d) The case in which the models of (a) and (b) are interpolated at the ratio of 1:1. This provides a baseline for the lack of access to the original data.

Additional Datasets We used two types of additional datasets. One is (e) all data in 2005 newspapers are used for adaptation (Normal Case). The other is (f) only the data unknown to the original model are used (Active Learning). In practical cases, we want to adapt the model when we find a failure of the original model. Therefore, case (f) is a practical setting. The additional datasets of the original models have different numbers of

Type	Method/Dataset	Transfer		Online	
		Test1	Test2	Test1	Test2
Baselines	(a) Original Model	**70.90%**	61.49%	**71.41%**	62.60%
	(b) Additional Only	56.73%	**75.66%**	57.07%	**76.36%**
	(c) Original + Additional Data	**70.99%**	**75.77%**	**72.00%**	**76.58%**
	(d) Interpolation	68.70%	72.28%	68.49%	72.98%
Model Adaptation	(e) Normal Case	64.26%	**75.78%**	66.87%	**75.78%**
	(f) Active Learning	50.32%	63.29%	57.28%	65.81%

Table 2: Test Set Accuracies of Methods and Datasets

entities, 34,950 and 33,633 for the Transfer and Online cases, respectively.

Tuning The hyperparameters are optimized against the development set when the original models are trained and the same values are used in all experiments.

3.2 Results of the Methods

The results are shown in Table 2.

First of all, focusing on baselines (a) and (b), Test set 1 yielded basically the highest accuracies for case (a), while for (b) it was Test set 2. Using datasets that are near to the test sets yields better model training in this task.

Focusing on case (c), in which training uses both the original and the additional datasets, the advantages of cases (a) and (b) are secured. However, although we applied domain adaptation, the improvements from cases (a) and (b) were little. This result indicates that the size of the additional dataset was sufficient and that the model matched the upper bound by using just the additional dataset. In addition, we confirmed that the accuracies of interpolation (d) were intermediate between those of (a) and (b).

While accuracy slightly differed with the learning method, the Transfer and Online cases exhibited the same tendency.

Next, for normal case (e) in model adaptation, both Transfer and Online achieved basically the highest accuracies against Test set 2. This result shows that model adaptation worked effectively. On the other hand, focusing on the accuracies of Test set 1, Online learning exhibited a smaller degradation from the original model (a) than Transfer. We suppose that this difference is due to the difference between maximum entropy and SCW-I, rather than that between the transfer/online learning. The maximum entropy method optimizes parameters based on the maximum a posteriori (MAP), and it is sensitive to probability distribution. In contrast, SCW-I used in Online is based on margin criteria, and ignores data outside the margin. Therefore, Online yielded smaller degradation.

In the case of active learning (f), the effects of model adaptation were little compared to the other cases. Namely, improvements against Test set 2 were slight and the accuracies of Test set 1 were degraded from the original model (a). Because transfer learning assumes that the target domain should be similar to the source domain, the dataset difference impacts performance significantly. We can conclude that we should collect (and use) all data for model adaptation regardless of whether or not the original model can correctly classify it.

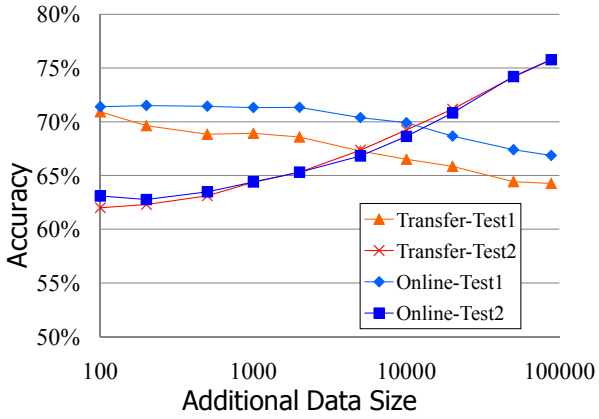
3.3 Accuracies according to Additional Data Size

Figure 4 plots accuracy versus the size of the additional datasets. In the normal case (e), the accuracies of Test set 2 improved with both the Transfer and the Online cases along with dataset size. In contrast, the accuracy of Test set 1 with Transfer degraded faster than Online, as described in Section 3.2. The degradation with Online started when over 2,000 data points were added. This result shows that the SCW-I algorithm of Online is relatively robust and remembers the previously learnt data.

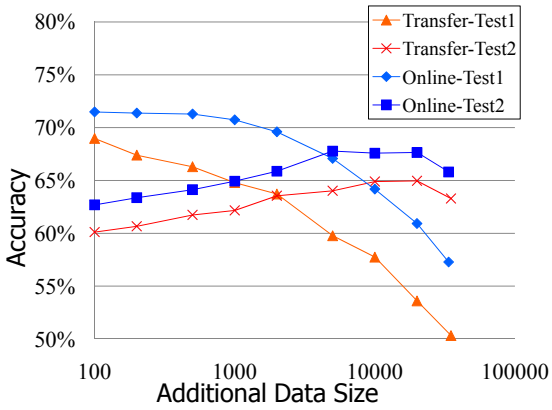
Focusing on active learning (f), the accuracies of Test set 2 were degraded with both Transfer and Online when all additional data was used. The addition of huge amounts of erroneous data causes a harmful effect regardless of the learning method used.

3.4 Hyperparameters in Transfer Learning

Finally, Table 3 shows the accuracies when the hyperparameters for the existing features in the original model and the new features that appear only in the additional data were distinguished by the method described in Section 2.2.2. Here, hyperparameter C_{OR} was set when the original model



(e) Model Adaptation (Normal Case)



(f) Model Adaptation (Active Learning)

Figure 4: Test Set Accuracy versus Size of Additional Data

Method	C_{OR}	C_{AD}	Test 1	Test 2
(e) Normal Case	0.1	0.001	70.33%	67.26%
	0.1	0.01	68.09%	72.71%
	0.1	0.1	64.26%	75.78%
(f) Active Learning	0.1	0.001	67.70%	65.15%
	0.1	0.01	61.28%	66.87%
	0.1	0.1	50.32%	63.29%

Table 3: Accuracies of Different Hyperparameters

was trained, and only C_{AD} was changed.

In the normal case, while the changes to the existing parameters were suppressed (small C_{AD}), the accuracy of Test set 2 decreased. However, it was higher than that of the original model (61.49% $\rightarrow$ 67.26%), and the accuracy on Test set 1 was almost constant (70.90% $\rightarrow$ 70.33%). If we have to adapt the model under the condition that the original performance is to be maintained, the two hyperparameter approach is effective.

In the active learning case, although we distinguished the hyperparameters, the results were not improved from the normal case.

4 Conclusions

We investigated the characteristics of model adaptation wherein the original training data cannot be accessed. We tested transfer learning (regularized adaptation) on the maximum entropy classifier and online learning (soft confidence-weighted learning). Our results are summarized as follows.

- If the additional dataset contains all data, regardless of whether it can be correctly classified by the original model or not, both transfer learning and online learning basically achieved the highest accuracy.
- However, the maximum entropy classifier with regularized adaptation changed more data, which the original model correctly classified, yielding more errors than online learning by SCW-I.
- Restricting the additional data to the data that the original model could not classify correctly had negative effects in our problem setting (i.e., the original dataset cannot be accessed).
- We could slightly adapt the model while retaining previous classification performance by distinguishing the hyperparameters for the existing features and those for the new features.

In natural language processing, structured learning is frequently used for sequential labeling, parsing, and so on. Our future work is to apply model adaptation to structured learning.

References

Adam L. Berger, Stephan A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Stanley F. Chen and Ronald Rosenfeld. 2000. A survey of smoothing techniques for maximum entropy models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50.

Koby Crammer, Mark Dredze, and Alex Kulesza. 2009a. Multi-class confidence weighted algorithms. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 496–504, Singapore.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009b. Adaptive regularization of weight vectors.

In *Advances in Neural Information Processing Systems 22 (NLPS)*, pages 414–422.

Hal Daumé, III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 256–263, Prague, Czech Republic.

Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of the 25th International Conference on Machine learning (ICML '08)*, pages 264–271, New York, NY, USA. ACM.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 109–117.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, pages 132–139, Prague, Czech Republic.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, 22(10):1345–1359, October.

Fei Sha and Brian Kingsbury. 2012. Domain adaptaion in machine learning and speech processing. Interspeech 2012 Tutotial. http://www-bcf.usc.edu/˜feisha/pubs/IS2012Tutorial.pdf.

Jialei Wang, Peilin Zhao, and Steven C. Hoi. 2012. Exact soft confidence-weighted learning. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 121–128, New York, NY, USA. ACM.

Li Xiao and Jeff Bilmes. 2006. Regularized adaptation of discriminative classifiers. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing Volume I*, pages 237–240.