Semi-Supervised Learning for Blog Classification

Daisuke Ikeda*
Department of Computational
Intelligence and Systems Science,
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku,
Yokohama, Kanagawa, Japan

Hiroya Takamura and **Manabu Okumura**
Precision and Intelligence Laboratory,
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku,
Yokohama, Kanagawa, Japan

Abstract

Blog classification (e.g., identifying bloggers' gender or age) is one of the most interesting current problems in blog analysis. Although this problem is usually solved by applying supervised learning techniques, the large labeled dataset required for training is not always available. In contrast, unlabeled blogs can easily be collected from the web. Therefore, a semi-supervised learning method for blog classification, effectively using unlabeled data, is proposed. In this method, entries from the same blog are assumed to have the same characteristics. With this assumption, the proposed method captures the characteristics of each blog, such as writing style and topic, and uses these characteristics to improve the classification accuracy.

INTRODUCTION

With the rapid growth of blogs, their value as source of information is increasing. A huge amount of work has been devoted to natural language processing (NLP) that handles blogs. In particular, the blog classification is one of the most well-studied problems (e.g., bloggers' gender or age identification, topic categorization). These attributes of blogs are going to be important, for example, when data mining in blogs is applied to examine people's behavior, such as determining what kind of people buy what products, and when.

This classification problem is usually solved by applying supervised learning techniques, because supervised learning is one of the most promising methods in the field of text classification. Although, obtaining high performance requires a large amount of labeled data, such manually labeled blogs are not always available. In contrast, unlabeled blogs can easily be collected from the Internet. Hence, semi-supervised learning methods using unlabeled data to improve classification performance are highly appropriate for blog classification.

The characteristics of a blog, e.g., its writing style or topics, should be important information source for some blog classification tasks. To obtain high performance, we have to take such characteristics into consideration..

Therefore, this paper proposes a semi-supervised blog classification method that captures the characteristics of blogs. The proposed method uses a huge number of unlabeled blogs to extract useful features. We tested our method on two tasks: bloggers' gender classification and age classification. In addition to describing our method in this paper, we also report the results of these experiments.

RELATED WORK

Blogs are usually published in a timely manner and contain people's opinions or impressions. Because of these features, there has been much work on using blogs to obtain trends, opinions, or sentiment on a variety of things.

In relation to this work, the problem of classifying blogs has recently been studied, including spam filtering (Kolari, Finin, & Joshi 2006) and identification of bloggers' age, gender (Yasuda *et al.* 2006; Schler *et al.* 2006). By running such classification as an initial process in any blog analysis, we can filter out blogs that are not the target of the analysis. We can also conduct trend or sentiment analysis separately for each age group or gender. A few services putting such technologies into practical use have already been launched [1], and through these kinds of applications, blog analysis technologies will be widely disseminated.

Most of the previous work has treated blog classification tasks as a supervised learning problem, as we described above. Namely, they applied text classification methods by regarding an input blog as a long text block that is the concatenation of all entries in the blog. To take advantage of a huge number of unlabeled blogs on the web, we need to use semi-supervised learning techniques although supervised learning techniques are well-studied methods.

There has been much work on semi-supervised text classification, including EM algorithms (Nigam *et al.* 2000) and co-training (Blum & Mitchell 1998). Although these methods can be applied in blog classification, they are designed for text classification. The method proposed in this paper is specifically designed for blog classification.

The proposed method can be regarded as an instance of Alternating Structure Optimization (ASO) proposed by Ando et al. (2005a). ASO is a machine learning frame-

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

*Currently with Google Japan, Inc., 6F Cerulean Tower, 26-1 Sakuragaoka-cho, Shibuya-ku, Tokyo.

[1] blogWatcher `http://blogwatcher.pi.titech.ac.jp/`, blogeye `http://blogeye.jp/`

work for multi-task or semi-supervised learning. It is effective on a variety of NLP tasks including text classification (Ando & Zhang 2005a), word sense disambiguation (Ando 2006), part-of-speech tagging (Ando & Zhang 2005b), semantic role labeling (Liu & Ng 2007), and domain adaptation (Blitzer, McDonald, & Pereira 2006; Blitzer, Dredze, & Pereira 2007). The details of the ASO algorithm and its relation to the proposed method are described later.

SEMI-SUPERVISED LEARNING FOR BLOG CLASSIFICATION

Sub-Classifiers

As we described above, the characteristics of blogs can provide valuable information for classifying of blogs. Most of the previous work, however, did not consider such information in building blog classifiers, because these characteristics are hard to represent as vectors, such as bag-of-words in a straightforward way. If we can represent these characteristics as vectors, we can make good use of these characteristics by applying existing classifiers with those features.

In this paper, we represent the characteristics of blogs in terms of relative similarity to other blogs, e.g., “blog A is dissimilar to blog B and is similar to blog C”. Information on which blogs are similar to a blog and which blogs are dissimilar to the blog can be considered as a kind of representation of the blog’s characteristics. Indeed, two blogs that are similar to the same blog would also be similar to each other. To implement our idea, we require a model measuring how similar the input blog is to each of the other blogs. By only looking at two blogs, however, it is difficult to define in what situation they should be judged as similar. We also have to know what sort of blogs exist besides those two blogs. For example, a blog about football and a blog about baseball should be similar to each other as compared with other pairs of blogs in an entire blog dataset. They should be regarded as dissimilar, however, when only sports-related blogs are considered.

Therefore, instead of the absolute similarity, we use the relative similarity with respect to two blogs, e.g., “blog A is similar to blog B rather than to blog C”. This is accomplished by using a classifier that, given an input blog and a pair of two blogs, outputs which of the two blogs the input blog is more similar to. We call such a blog pair a *reference (blog) pairs*. We use a linear discriminative model $score(\boldsymbol{x}) = \boldsymbol{u} \cdot \boldsymbol{x}$, where $\boldsymbol{x}$ is a vector representation of a blog, such as a bag-of-words vector. The sign of $score(\boldsymbol{x})$ indicates which of the reference pair the input blog is more similar to. Its absolute value indicates the degree of similarity. The vector $\boldsymbol{u}$ is a model parameter estimated with a reference pair composing the model. We describe the details of the learning procedure in the following section. We call this classifier the *sub-classifier* because its role is to help construct the classifier for the target task. Similarly, we call the classifier for the target task the *main classifier*.

Since sub-classifiers concern only characteristics of blogs such as writing styles, they are independent of the target task. This means that sub-classifiers can be learned with unlabeled blogs and constructed in huge number because of the ease of collecting unlabeled blogs. By considering similarities with respect to a huge number of reference blogs, we can capture characteristics of target blogs more precisely.

Training Sub-Classifiers

Next, we explain how we train sub-classifiers.

Usual supervised learning approaches cannot be applied to sub-classifiers because of the lack of training data with labels indicating which of the reference blog pair the target blog is more similar to. Therefore we focus on a special property of blogs, namely, that they consist of multiple entries. We regard each entry in each of the reference blog pairs as a training example and then train the model accordingly. Specifically, we regard the entries in one blog of the reference pair as positive examples, and the entries in the other as negative examples.

With this setting, a large number of training examples can be obtained. Furthermore, when blogs are collected from the web, it is usually determined which entry belongs to which blog. This means that we have absolutely no need for manually labeled reference blogs, and that we can train the model from any two blogs collected from the web.

The classifier trained in this method can be seen as a model classifying input entries into two classes. Here, we assume that all entries in one blog have the same characteristics, such as writing styles. Different blogs are usually maintained by different people and vary in their characteristics. Meanwhile, each blog usually consists of a set of entries written by one person, which are likely to share the same characteristics. Therefore, sub-classifiers are expected to capture the differences in characteristics of the reference blogs. By classifying blogs with these sub-classifiers, we can measure which of the reference blog pair the input blog is more similar to.

Why We Train Sub-Classifiers

In the previous section, we explained how to train sub-classifiers. If we simply want to measure relative similarities to reference blog pairs, however, we might not have to conduct learning. For example, we could use the cosine similarity or inner product commonly used in the area of information retrieval, in order to measure the similarities to reference blogs, and transform these measures to relative values. These values could be also seen as a kind of relative similarity reflecting the characteristics of blogs.

In contrast, the proposed learning procedure allows us to capture the differences in characteristics between the reference blogs used to build the sub-classifier.

For instance, let us consider a sub-classifier built from a blog about football and a blog about baseball. These two reference blogs have much in common, because they are both about sports. Therefore, when we use the similarity without learning, the relative similarity would be neutral in most cases. It would be hard to determine which reference blog the input is more similar to. By contrast, the proposed model assigns large weights to features that are effective for discriminating two reference blogs. In the above example, features that frequently occur only in one of the two reference blogs such as “bat” or “offside”, are expected to have large weights. Because of these heavily weighted features, we can clearly resolve which reference blog the input blog is more similar to.

Semi-Supervised Learning with Sub-Classifiers

Next, we propose a semi-supervised learning method for blog classification with the sub-classifiers introduced above.

The output of each sub-classifier shows which reference blog the input blog is more similar to. This output can be regarded as a numerical representation of the characteristics of the input blog from one viewpoint. By building a huge number of sub-classifiers from many unlabeled blogs, we can obtain the characteristics of the input blog from many different viewpoints. Thus, a vector whose elements are the outputs of the sub-classifiers can be seen as a feature vector capturing the characteristics of the input blog. We use this vector as additional features to the main classifier. Therefore, the input vector to the main classifier will have as many additional dimensions as the number of sub-classifiers.

This method can be seen as a semi-supervised learning method using information from unlabeled data to represent the characteristics of blogs as a vector.

The learning procedure is formulated as follows. Given K sub-classifiers (denoted by $\boldsymbol{u}_0, ..., \boldsymbol{u}_{K-1}$), we construct a matrix $U \equiv [\boldsymbol{u}_0, ..., \boldsymbol{u}_{K-1}]$. The vector whose elements are the outputs of the sub-classifiers can be represented as $U^T\boldsymbol{x}$. We concatenate this vector with the original input vector $\boldsymbol{x}$. Then, the classifier we need is trained from training examples: $\left\{ \left[\boldsymbol{x}_i^L, U^T\boldsymbol{x}_i^L\right], y_i \right\} (i = 1...N)$.

Also for the main classifier, we use a linear discriminative classifier: $y = sign(\boldsymbol{w} \cdot \left[\boldsymbol{x}, U^T\boldsymbol{x}\right])$.

In the classification phase, the sub-classifiers are applied to the input, and then, the input vector is extended with the outputs of the sub-classifiers, as in the training phase. Hence, the complexity of this procedure is directly proportional to the number of sub-classifiers. Since the matrix U is independent of the input blog, however, we can calculate $\boldsymbol{w}U^T$ in advance so that the complexity should become constant with respect to the number of sub-classifiers.

Most of the learning algorithms assign a weight to each feature in the input vector in the training phase. The weights are used for calculating classification scores in the classification phase. The weight for a feature added by the proposed method indicates the correlation between the corresponding sub-classifier and the target problem. For example, if we are examining bloggers' gender classification, a sub-classifier built from two reference blogs whose authors' genders are different, will have a large weight.

Figure 1 summarizes the proposed algorithm.

ALTERNATING STRUCTURE OPTIMIZATION

As we mentioned above, the proposed method can be regarded as an instance of Alternating Structure Optimization (ASO) proposed by Ando et al (2005a). This section describes the details the ASO algorithm and the relation between the proposed method and the ASO algorithm.

```
Input: Labeled blogs $\{(\boldsymbol{x}_i^L, y_i)\}(i = 0, ..., N-1)$;
    unlabeled blogs $\{\boldsymbol{x}_i^U\}(i = 0, ..., M-1)$.
    Each blog $\boldsymbol{x}$ contains some entries $\{\boldsymbol{e}|\boldsymbol{e} \in \boldsymbol{x}\}$.
Parameter: Number of sub-classifiers, $K$.
Output: Parameter vector of the model, $\boldsymbol{w}$; matrix $U$ .
FOR $l = 0$ TO $K-1$
  Select a pair of two unlabeled blogs
    randomly (denoted by $\boldsymbol{x}_s^U$, $\boldsymbol{x}_t^U$).
  Find the parameter vector $\boldsymbol{u}_l$ of the sub-classifier,
    which discriminates $\boldsymbol{e} \in \boldsymbol{x}_s^U$ and $\boldsymbol{e} \in \boldsymbol{x}_t^U$.
NEXT
$U \equiv [\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{K-1}]$
Train $\boldsymbol{w}$ from the training examples,
    $\left\{ \left[\boldsymbol{x}_i^L, U^T\boldsymbol{x}_i^L\right], y_i \right\} (i = 0, ..., N-1)$ .
```

Figure 1: Proposed algorithm

Outline of ASO Algorithm

ASO is a machine learning framework for semi-supervised learning and multi-task learning. It has been reported effective for several applications in natural language processing, including text classification and part-of-speech tagging.

The ASO algorithm requires using *auxiliary problems*, which are problems other than the target problem. The effectiveness of the ASO algorithm heavily depends on the selection of auxiliary problems. Ando et al. (2005a) proposed the following auxiliary problems for text classification and text chunking.

- Prediction of frequent words: This auxiliary problem was proposed for text categorization according to topics. A set of content words is split into two subsets. The objective of this problem is to predict the word occurring most frequently among the words in one subset by using only the words in the other subset.
- Prediction of words in sequence: This auxiliary problem was proposed for text chunking, e.g., part-of-speech tagging. In this problem, the word occurring at each position is predicted from its context.

Auxiliary problems of this type are solved as a large number of binary classification problems. In the case of predicting frequent words, we can train as many classifiers as the size of the vocabulary, for instance. In addition, since these auxiliary problems can be trained without labeled data for the target problem, we can use unlabeled data for training.

By simultaneously training these many auxiliary problems in the manner of multi-task learning, information that is usually hard to take into consideration, such as the relations between features, is extracted. This information is used to train a main classifier.

The training procedure of ASO is as follows.

1. For each binary classification problem introduced by the auxiliary problems, learn $\boldsymbol{u}_i$ from unlabeled data.
2. Define a matrix $U \equiv [\boldsymbol{u}_0, \boldsymbol{u}_1, ...]$.
3. Decompose the matrix U into the form $U = V_1DV_2^T$ by applying the singular value decomposition (SVD), and store the first h columns of $V1$ as rows of Θ.

4. Transform the training data $\{\boldsymbol{x}_i, y_i\}(i = 0, ..., N-1)$ into $\left\{\left[\boldsymbol{x}_i, \Theta^T \boldsymbol{x}_i\right], y_i\right\}$ $(i = 0, ..., N-1)$ and find the parameter vector of the target problem by using the transformed training examples.

Although Ando et al. provided a theoretical justification for transforming of the matrix by applying the SVD, we did not use such transformation in our experiments, because we did not obtain significant improvement in classification accuracy with this approach in our preliminary experiments.

Relation to Proposed Method

The proposed method can be seen as a novel application of ASO for blog classification. From this point of view, our contribution is that we explored a new auxiliary problem, i.e., a sub-classifier, which is suitable for blog classification.

Ando et al. mentioned that semi-supervised learning with the ASO algorithm requires auxiliary problems with the following two characteristics:

- Automatic labeling: Labeled data for the auxiliary problems can be automatically generated from unlabeled data.
- Relevancy: Auxiliary problems should be related to the target problem to some degree.

Note that "two problems are related" means that one problem is at least somewhat informative for solving the other.

Sub-classifiers that we proposed also meet these two conditions and can be regarded as an auxiliary problem.

- Automatic Labeling: When blogs are collected from the web, information on which entry belongs to which blog is determined. From this information, we can automatically generate labeled examples of the sub-classifier.
- Relevancy: One of our first motivations is that blog characteristics that can be captured by sub-classifiers provide valuable information when classifying blogs. When randomly selected reference blogs respectively correspond to different classes of the target problem, the sub-classifier will indicate high relevancy to the target problem. For example, a sub-classifier built from two reference blogs whose authors' genders are different may be informative for gender classification of bloggers.

Applications to Other Tasks

Most auxiliary problems proposed so far capture relations between features, by predicting a part of the feature set from other features. In contrast, our sub-classifier learns the relation or difference between sets of unlabeled data and is thus different from existing auxiliary problems. This is enabled by the properties of blogs: *blogs consist of several entries*, and *the entries in one blog have the same characteristics*.

Applying an auxiliary problem of this type to other tasks requires meeting the following conditions.

- Examples are divisible: one example has to be divisible into several pieces. In our auxiliary problem, we divide blogs into entries.
- The divided examples partly share characteristics: without such characteristics, we cannot solve the classification problems of predicting which piece is from which example. In our case, we expect that the entries in one blog have the same characteristics, such as writing style.
- The shared characteristics are related to the target problem: this condition is identical to the "relevancy" of an auxiliary problem. In our auxiliary problem, the characteristics of blogs are informative for classification.

EXPERIMENTS

We conducted several experiments to show the effectiveness of the proposed method. As target problems, bloggers' gender identification and age identification were bothe examined. The goal of gender identification was to classify blogs into the male class and the female class. The goal of age identification was to classify blogs into five classes, from the teens to the fifties. Since these problems are the most basic problems in blogger attribute identification, some work on these problems has already been done (Schler *et al.* 2006).

Datasets

Almost the same datasets were used in both experiments. We collected Yahoo! blogs[2] whose authors' genders or ages were given in their profiles and used them as labeled blogs. Unlabeled blogs were collected from Livedoor[3]. From this set of unlabeled blogs, reference blogs were randomly sampled and used for training sub-classifiers.

Note that, the distribution of user population or community depends on the service provider. Therefore, to obtain better classification performance, both the labeled and unlabeled data should actually be collected from the same source, i.e., the same blog service providers. However, it is impractical to prepare an unlabeled dataset for every service provider. Restricting the use of unlabeled blogs to a specific provider would ruin the blogs' advantage of being easily collected in numbers. For this reason, we evaluated our method with labeled and unlabeled blogs from different sources.

Experimental Setting

We used a perceptron-based online learning algorithm (Crammer *et al.* 2006) to train the sub-classifiers, because we needed to train a tremendous number of classifiers, and this algorithm is known to be fast and effective. As the training algorithm for the target problem, we used Support Vector Machines (SVM), which is well known and commonly used in the area of natural language processing because of its performance. The SVM implementation that we used was TinySVM [4]. The SVM hyperparameters, including the soft margin parameter, were set to the default settings. We used bag-of-words as features for all classifiers, i.e., the sub-classifiers, the gender classifier, and the age classifier. Hence, in the proposed method, each input vector of the target task is a concatenation of the bag-of-words vector and the real-value vector whose elements are the outputs of the

[2] http://blogs.yahoo.co.jp/

[3] http://blog.livedoor.com/

[4] http://chasen.org/~taku/software/TinySVM/

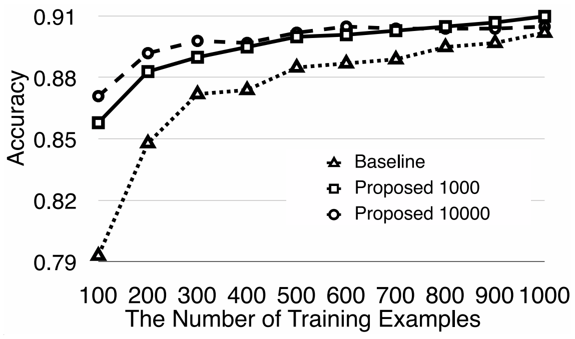

Figure 2: Results for bloggers' gender identification

Table 1: Results for bloggers' age identification

	One-versus-rest	Pairwise
Baseline	0.629	0.627
Proposed 1000	0.636	**0.642**
Proposed 10000	0.616	0.635

sub-classifiers. We used a linear kernel for all the experiments. The performance was evaluated in terms of the classification accuracy. We compared the following methods:

- **Baseline**: Ordinary supervised learning, making no use of unlabeled blogs.
- **Proposed 1000**: The proposed method, using 1,000 sub-classifiers to extend the input vector.
- **Proposed 10000**: The proposed method, using 10,000 sub-classifiers.

Gender Identification

First, we report the results for bloggers' gender identification. We used 2112 examples for testing, while the number of training examples ranged from 100 to 1000.

Figure 2 shows the results of the experiments. Both Proposed 1000 and Proposed 10000 achieved better performance than the baseline did, regardless of the number of training examples. The improvement was remarkable when we used only a limited amount of training data. Thus, we conclude that the proposed method works well even when collecting many labeled blogs for the target problem is hard.

In addition, Proposed 10000 tended to achieve greater accuracy than Proposed 1000, especially with a small training dataset. If we had more unlabeled data, the classification accuracy would be further improved.

Age Identification

We used 2000 blogs for testing and 1314 blogs for training.

Age identification is a five-class classification problem. Thus, we had to extend the SVM to the multi-class case. We applied both the *one-versus-rest* method and the *pairwise* method, which are well-known methods for applying binary classifiers to multi-class classification problems.

Table 2: Results for age identification for each class

	Baseline	Proposed 1000	Proposed 10000
10 vs 20	0.863	0.866	0.864
10 vs 30	0.893	0.919	0.917
10 vs 40	0.945	0.951	0.954
10 vs 50	0.946	0.943	0.946
20 vs 30	0.751	0.769	0.736
20 vs 40	0.908	0.912	0.908
20 vs 50	0.957	0.960	0.962
30 vs 40	0.790	0.792	0.747
30 vs 50	0.871	0.893	0.906
40 vs 50	0.703	0.703	0.734

Table 1 shows the results. Proposed 1000 outperformed the baseline in both the one-versus-rest and pairwise settings. This suggests that the proposed semi-supervised method also worked effectively in this task.

Both versions of the proposed methods obtained better results in the pairwise setting than in the one-versus-rest setting, whereas the baseline obtained a similar accuracy in both settings. The reason could be that the setting of the sub-classifiers, i.e., discrimination between two blogs, is similar to the pairwise setting, and hence the relevancy between the sub-classifiers and the target classifier should be high.

Proposed 10000 did not outperform Proposed 1000 in the pairwise setting, and it did not outperform even the baseline in the one-versus-rest setting. To clarify the reason for this, we examined the classification results for each class. Table 2 shows the age identification results for each pair of classes. The leftmost column shows the target classes for each pair of classes. Since we used the pairwise setting here, each classification had two classes. For example, "10s vs 30s" means the blogs were classified into the teens class or the thirties class. Note that, we evaluated the methods in terms of the classification accuracy, and only the test examples belonging to one of the two classes in each row were used. In the row of "10s vs 30s", for example, only the blogs whose authors were in their teens or thirties were used as test examples.

From these results, we observed that Proposed 10000 obtained low accuracies for "20s vs 30s" and "30s vs 40s". This was because these pairs of classes are hard to distinguish using blog characteristics, such as writing style. It would be difficult even for humans to classify blogs into one of the twenties and thirties classes using only blog characteristics. The proposed method captures the characteristics of blogs. Therefore, for classification into two classes with similar characteristics, as in this case, we cannot expect high performance if the number of sub-classifiers and their influence are large. This problem can be avoided by introducing a parameter controlling the influence of unlabeled data. Even with such difficult problems, the proposed method outperformed the baseline when the training dataset is very small. In the case of "20s vs 30s" with 100 training examples, both Proposed 1000 and Proposed 10000 obtained an accuracy of 0.700, while the baseline obtained 0.669.

DISCUSSION

Compared with existing semi-supervised learning methods, the proposed method has the advantages listed below.

- The proposed method can be combined with existing methods, because it merely extends input vectors. Thus, EM algorithms or co-trainings are easily applicable at the same time, giving the same extension to unlabeled data.
- The proposed method is relatively fast and consumes less memory than do existing semi-supervised learning methods. This is because sub-classifiers are trained independently, and thus there is no need to handle all data simultaneously, requiring less memory. In addition, using fast training algorithms, such as online algorithms, enables us to reduce the computational time.
- The proposed method can use a variety of features in unlabeled data for training the sub-classifiers, e.g., n-grams, or even face marks. Furthermore, while it will take a long training time, we can also use kernel methods to enable non-linear sub-classifiers.

Next, we discuss the disadvantages of our method and some ways to improve it.

- The proposed method can incorporate only a portion of the entire set of unlabeled blogs. We reluctantly limited the number of sub-classifiers and chose them randomly because of the huge number of the unlabeled blogs. This procedure means that some unlabeled data is randomly abandoned. We are trying to avoid this by exploring a new strategy for building and selecting the sub-classifiers.
- The proposed method cannot control the influence of unlabeled blogs. As we observed in the experiments, when the number of sub-classifiers was increased, the influence of unlabeled blogs seemed to be over-emphasized. One possible way to avoid this is to introduce an additional parameter controlling the influence of the sub-classifiers.

CONCLUSION

In this paper, we have proposed a semi-supervised blog classification method that captures the characteristics of blogs, such as writing style or topics. We evaluated the proposed method and demonstrated that it can improve the accuracy of both gender identification and age identification of bloggers. In contrast to existing methods, the proposed method focuses on two properties of blogs, namely, that *blogs consist of several entries*, and that *the entries in one blog should have the same or, at least, similar characteristics*.

For our future work, we plan to explore a new strategy for building and selecting sub-classifiers. Since, we used a simple strategy in this paper, i.e., to randomly selecting pairs of blogs, there is room for improvement through concepts such as developing effective sub-classifier selection methods. For example, in building a sub-classifier, we can involve three or more blogs or only one blog by using a multi-class classifier or a one-class classifier, respectively, instead of a binary classifier. In addition, we require feature engineering for sub-classifiers. Fo capturing writing style, there should be a more effective feature set than the bag-of-words approach used here. We would also like to combine our method with existing semi-supervised learning methods for text classification, such as EM algorithms or co-training. Experiments on other blog classification tasks, such as spam filtering and topic categorization, will also be included in our future work (Crammer *et al.* 2006).

References

Ando, R. K., and Zhang, T. 2005a. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* Vol 6:pp.1817–1853.

Ando, R. K., and Zhang, T. 2005b. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, pp.1–9.

Ando, R. K. 2006. Applying alternating structure optimization to word sense disambiguation. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-2006)*, pp.77–84.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes, and blenders: Domain adaptation for sentiment classification. In *Proceedings of Association of Computational Linguistics (ACL-2007)*, pp.440–447.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP-2006)*, pp.1166–1169.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory (COLT-1998)*, pp. 92–100.

Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; and Singer, Y. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* Vol. 7:pp.551–585.

Kolari, P.; Finin, T.; and Joshi, A. 2006. SVMs for the blogosphere: Blog identification and splog detection. In *Proceedings of AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, pp.92–99.

Liu, C., and Ng, H. T. 2007. Learning predictive structures for semantic role labeling of nombank. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, pp.208–215.

Nigam, K.; McCallum, A.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning* Vol .39(No.2):pp. 103–134.

Schler, J.; Koppel, M.; Argamon, S.; and Pennebaker, J. 2006. Effects of age and gender on blogging. In *Proceedings of AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, pp.199–205.

Yasuda, N.; Hirao, T.; Suzuki, J.; and Isozaki, H. 2006. Identifying bloggers' residential areas. In *Proceedings of AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, pp.231–236.