Asia Pacific Journal of Information Systems
Vol. 29 No. 4 (December 2019), 789-816

ISSN 2288-5404 (Print) / ISSN 2288-6818 (Online)
https://doi.org/10.14329/apjis.2019.29.4.789

Special Issue: Data Analytics in Artificial Intelligence Era

Text Classification with Heterogeneous Data Using Multiple Self-Training Classifiers

William Xiu Shun Wong[a], Donghoon Lee[b], Namgyu Kim[c,*]

[a] Senior Consultant, Biz Consulting Team, Datasolution Inc., Korea
[b] Staff, BI LAB, Cafe24 Corp., Korea
[c] Professor, School of Management Information Systems, Kookmin University, Korea

ABSTRACT

Text classification is a challenging task, especially when dealing with a huge amount of text data. The performance of a classification model can be varied depending on what type of words contained in the document corpus and what type of features generated for classification. Aside from proposing a new modified version of the existing algorithm or creating a new algorithm, we attempt to modify the use of data. The classifier performance is usually affected by the quality of learning data as the classifier is built based on these training data. We assume that the data from different domains might have different characteristics of noise, which can be utilized in the process of learning the classifier. Therefore, we attempt to enhance the robustness of the classifier by injecting the heterogeneous data artificially into the learning process in order to improve the classification accuracy. Semi-supervised approach was applied for utilizing the heterogeneous data in the process of learning the document classifier. However, the performance of document classifier might be degraded by the unlabeled data. Therefore, we further proposed an algorithm to extract only the documents that contribute to the accuracy improvement of the classifier.

Keywords: Text Mining, Text Classification, Heterogeneity Learning, Semi-Supervised Learning, Ensemble Learning

Ⅰ. Introduction

During the past few years, many efforts have been made to discover new value from vast amounts of data through data mining techniques such as classification, clustering, frequency analysis, and association analysis. Especially for classification analysis, data mining techniques has been used in various practical areas such as corporate bankruptcy prediction, repurchase prediction, and stock price prediction. However, rapid growth of internet and the recent popularization of smart deveices have gen-

*Corresponding Author. E-mail: ngkim@kookmin.ac.kr Tel: 8229105425

erated huge amounts of unstructured text data through different social media platforms such as news feed, blog, electronic mail, and microblog. It is more difficult to effectively perform classification analysis with the presence of unstructured text data by using traditional data mining methods.

Text mining has received increasing attention of many expects from different fields by reason of its ability to discover the knowledge from the data sources that contain unstructured or semi structured information. In order to automatically classify and identify the hidden patterns of text data from different sources, text mining can be used to deal with the different operations, such as data mining, Natural Language Processing (NLP), classification (supervised, unsupervised and semi-supervised), information retrieval, and machine learning techniques. Among these operations, we focus on text classification in this study.

The main purpose of text classification is assigning one or more pre-defined category labels to text documents. There are two types of text classification, which recognized as single label classification and multi-label classification. In details, single label classification only classifies a document to a single label, while multi-label classification may classify a document to more than one label. In this paper, single label text classification is our main focus. Various machine learning approaches available in the field of text classification such as Naive Bayes, K-Nearest Neighbor, Support Vector Machine (SVM), Artificial Neural Network, and Decision Tree. Most of the machine learning algorithms are based on supervised learning, which use labeled documents for learning. Precision, recall, and F1-Score are mainly applied for evaluating the performance of document classifier.

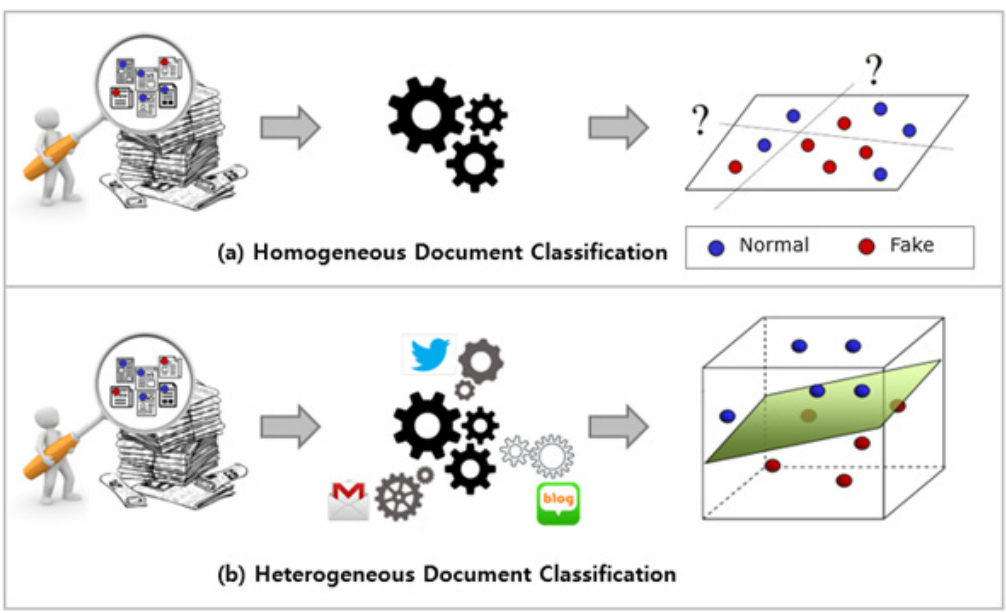
The performance of a classification model can be varied depending on what type of words contained in the document corpus and what type of features generated for classification. Therefore, many attempts have been actively conducted to improve the accuracy of text classification. Angelova and Weikum (2006) proposed a new graph-based classifier algorithm and achieve higher classification accuracy than prior work. While Mitra et al. (2007) presents a Latent Semantic Index (LSI) coefficient based Least Square Support Vector Machine module for text classification and the comparison result between linear SVM based classifiers and neural network based classifiers shows that the proposed classifiers improves text classification performance significantly. Many of these studies have attempted to improve the performance of text classification by proposing a new modified version of the existing algorithm or creating a new algorithm. It is hard to achieve further improvement by using this kind of approaches as such approaches already reached their limitation. Therefore, we attempt to modify the utilization method of learning data that required for constructing a classification model, rather than suggesting a new algorithm or modifying the existing algorithms.

Based on previous studies, the performance of the classifier is usually affected by the quality of training data at the time of which this classifier is constructed (Wu and Zhu, 2008). The real-world datasets usually contain noise and the classifiers are build based on these real-world data, which means there ais a big possibility the decisions generated by the classifiers will be affected. However, many attempts have been made to enhance the noise-robustness of the classifiers by manipulating with noisy data. Kim et al. (2011) investigate the noise tolerance ability of existing fault prediction methods by manually injecting noises. While Liu et al. (2015) proposed a cluster based feature selection method for software fault prediction with noise. By using both noise free

and noisy datasets, they also verify the robustness of the method on both datasets (Liu et al., 2015). Sáez et al. (2013) investigates how well that several multiple classifier systems behave with noisy data by conducting a huge experimental study by comparing the performance and the robustness towards noise of different multiple classifier systems. In our study, we assume that the data from different domains might have different characteristics of noise. In this study, we called those data as heterogeneous data and will be utilized in the classification process.

In order to construct the classifier, machine learning algorithm is performed based on the assumption that the characteristics of training data and target data are the same or very similar to each other (L'Heureux et al., 2017). If the machine-based classifier is applied in an environment where these assumptions are not followed, the accuracy of the classification are expected to be low. In the case of structured data, although there is a difference in value between learning data and the target data, but the attributes or features still remain the same. Different from structured data, the features of unstructured data such as text document are determined based on the vocabularies of the document. If the perspective of the learning data and target data are different, the features between these two data can be appeared differently. In this study, unlike the previous studies which minimize the negative influence of the noise on the text classification, the robustness of the classifier will be enhanced by injecting the noise artificially into the learning process and hence improving the classification accuracy. In other words, we extract features from heterogeneous data sources that have completely different characteristics from the original data source which also is our classification target. The heterogeneity learning will be performed by injecting these features into the process of learning the classifier as a kind of noise.

<Figure 1> shows an example of a homogeneous document classifier and a heterogeneous document classifier were constructed to classify the news articles

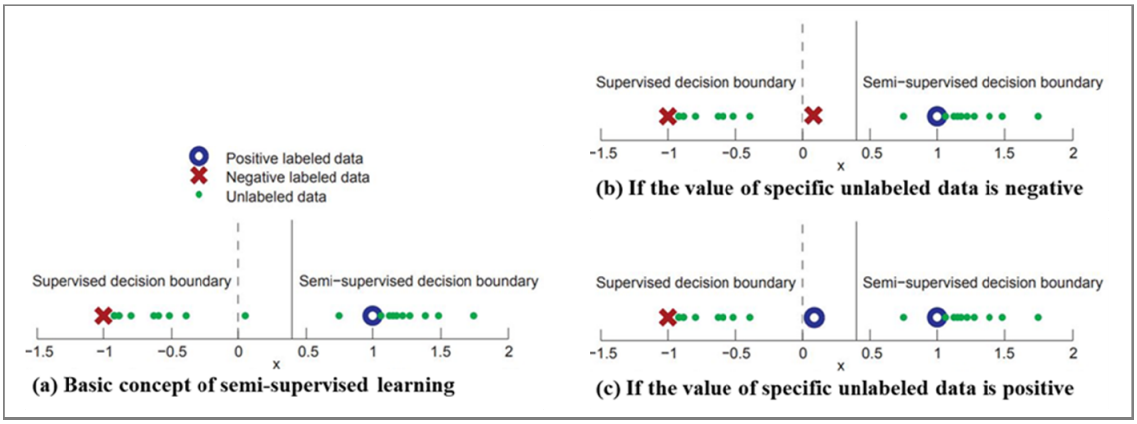


<Figure 1> Comparison between Two Different Document Classification

into their corresponding category. In <Figure 1(a)>, a classification model is build based on learning the news articles and was used to apply on news article classification. <Figure 1(b)> shows a classification model with the injection of the features extracted from heterogeneous data sources into the learning process of news articles. In this study, this process is called heterogeneity learning. As a result, the model is applied to the news article classification. For the traditional classifier (a), since the learning data and target data are the same as the news data, the classification criterion is limited to the features extracted from the news articles. Therefore, when a news article containing a new issue appears, it is difficult to classify it correctly based on the existing classification criterion. On the other hand, as the document classifier (b) performs heterogeneity learning by using not only news data, but also using various heterogeneous data such as blogs and tweets. Therefore, compared to document classifier (a), it is possible to have a more sophisticated classification criterion by performing the learning from various viewpoints.

In machine learning, a data processing step is often needed to convert data to adapt within a particular model. Nevertheless, these data seem to be formatted differently, because these data coming from various kind of sources. These cause difficulties for traditional machine learning algorithms because they are not developed to recognize different type of data representation at one time and to put them together in same generalization (L'Heureux et al., 2017). In order to perform heterogeneity learning by utilizing heterogeneous data in the process of learning the document classifier, a semi-supervised learning approach will be applied in our study. Semi-supervised learning (SSL) approach is getting attention from various machine learning fields, because it can overcome the limitation of supervised learning due to lack of classification data and able to benefit from unlabeled samples together with labeled ones (Zhu, 2008). Self-training is an effective SSL technique that able to solve the problems of insufficient amounts of labeled training examples and has been applied in several real-world practice fields (Li and Zhou, 2005). The self-training classifier model accepts that the predictions by its own are aim to be correct and does not assign any specific assumptions for the input data. In this study, we attempt to perform heterogeneity learning using this self-training approach. Specifically, we try to perform the heterogeneity learning by using the original data that used for learning as labeled data and applying the heterogeneous data as unlabeled data in the self-training process.

However, SSL classification not always give outperform result if compared to supervised learning classification. In <Figure 2(a)>, there are three kind of data, green dot represents unlabeled data, O represents positive labeled data, and X represents negative labeled data. When the classification criterion is set by using only two labeled data: O and X labeled data, the decision boundary is formed as $x = 0$ as shown by the dotted line. If a large number of unlabeled data, which shown as green dots in the figure were added for the learning, the decision boundary will be different. Although the correct class labels for these unlabeled data are unknown, but this unlabeled data gives more information to the learning and yet they form two groups. Specifically, it seems that the two labeled data are not the only one who contributes to the learning process. Since the classification criterion is affected by the unlabeled data, the decision boundary is changed as $x \approx 0.4$ and shown by the solid line. In other word, it can be seen that the specific unlabeled data between the dotted line and solid line is classified by supervised



<Figure 2> Basic Concept of Semi-supervised Learning (Zhu and Goldberg, 2009)

classifier as positive and semi-supervised classifier as negative. According to <Figure 2(b)>, if the original value of specific unlabeled data is negative, then the classification accuracy was improved by semi-supervised learning. On the other hand, if the original value of specific unlabeled data is positive as shown in <Figure 2(c)>, then the classification accuracy was getting worse by semi-supervised learning.

Nigam et al. (2000) state that classifiers sometimes display performance degradation and they suggest several possible sources of difficulties: numerical problems in the learning algorithm, mismatches between the natural cluster in feature space and the actual labels. Bruce (2001) used labeled and unlabeled data to learn Bayesian network classifiers. However, the Naive Bayes classifier displayed bad classification performance, and in fact the performance degraded as more unlabeled data were used. Cozman et al. (2003) also shows that the performance of a classifier can be degraded by the unlabeled data when there are conflicts between modeling assumptions used to construct the classifier and the actual model that creates data. Therefore, in this study, a method is proposed by extracting only the documents that contributing to the improvement of classification accuracy.

In this study, there are two main points to be emphasized: (1) Heterogeneity learning is performed by extracting the features not only from news, but also from heterogeneous data such as blogs, and twitter data. The extracted features will be injected into the classification learning process. (2) A method of extracting and applying only the classification rules that contributing to the accuracy improvement of a document classifier.

At first, aside from news data, two different heterogeneous data such as blog data and twitter data were used in the process of constructing the semi-supervised classifier. In this study, the semi-supervised classifier with heterogeneity injection will be called as heterogeneity classifier. In addition, in order to figure out the difference of using the homogeneous data and heterogeneous data in the process of constructing semi-supervised classifier, the news data without the labels will also be used together as homogeneous data. As a result, one homogeneous classifier (news data) and two different types of heterogeneity classifiers (blog data and twitter data) were generated. These three classifiers will be used to score the target documents, respectively. The scored results of these

three classifiers will be combined and the best scoring result will be selected for the next process.

Secondly, a method called Ensemble-Based Rule Selection Algorithm (ERSA) was proposed to select only the documents that contributing to the accuracy improvement of the classifier. As the semi-supervised approach not always guarantee outperform result compared to supervised approach, therefore we decide to include both scored results of supervised classifier and heterogeneity classifier in this study. Before applying the ERSA, the scored results of these two different classifiers were derived and combined. By using the ERSA, the ensemble of supervised classifier and heterogeneity classifier will select the most confidently classification rules to be applied for the final decision making.

The other sections of this paper will be further explained accordingly. Related work will be explained in Section 2, which introduces related studies on data heterogeneity, semi-supervised learning, and ensemble learning. Section 3 describes the detailed process of the proposed methodology. Further, Section 4 explains the experimental settings in details and also presents the empirical results on a real-world dataset. In Section 5, the conclusion of this paper will be described in the following sequences, which are contributions, limitations, and future works.

Ⅱ. Related Work

2.1. Data Heterogeneity and Robustness

In recent years, the amount of data is increasing at an extraordinary speed as the latest technological developments is growing rapidly in Web technologies, mobile services, and social media. By giving an example in social media, Twitter processes over 500 million tweets per day, these tweets were generated by 100 million daily active users in 2018 (Aslam, 2018). However, the conventional approaches are facing difficulties in seeking a way to deal with these tremendous data.

Garner has defined the concept of big data into high velocity, high volume, and high variety data. These big data require new analysis approaches to disclose the valuable insight, improve decision making, and optimize the process (Beyer and Laney, 2012). Big data is described as data that contains greater variety arriving in increasing volumes and with ever-higher velocity, which is also known as the three Vs. Volume refers to the quantity of generated and stored data. The size of the collected data will determine the value and potential insight. Velocity denotes both the rate in which data is received and the time frame in which they must be acted upon. Variety refers to heterogeneity of data types, representation, and semantic interpretation. Traditional data types were structured and fit neatly in a relational database. However, big data are often obtained from different sources and represent information from different sub-populations. As a result, big data are highly heterogeneous, especially when the data come in new unstructured data types (unstructured and semi-structured data types). In this study, we focus on the variety part of Big Data that includes the data heterogeneity.

The data analytics nowadays usually need to integrating and handling the diverse data from different sources, which these data can be diverse in terms of data format, data type, data model, data encoding, and semantics. These diverse data can be divided into three main heterogeneity categories: syntactic, semantic, and statistical heterogeneity (L'Heureux et al., 2017). Syntactic heterogeneity refers to diversity in data format, data type, data model, data encoding,

etc. On the other hand, semantic heterogeneity refers to differences in meanings and interpretations. Based on different parties of "world view", several datasets were developed. The semantic heterogeneity will be encountered if there is an attempt to combine those datasets. This causes problems for machine learning approaches because they are not designed to manage semantically diverse data. In addition, statistical heterogeneity indicates the differences in statistical properties among the different parts of an overall dataset.

In our study, we consider that heterogeneous text data is having similar characteristics with semantic heterogeneity and attempt to apply heterogeneous text data from different sources in text classification model learning process. In text classification, the features of text data are determined based on the vocabularies of the document. If the perspective of the learning data and target data are different, the features between these two data may appear differently. Therefore, in order to build a good classifier, we attempt to improve the features of the existing learning documents by applying the vocabularies that are extracted from heterogeneous documents of different data sources.

Furthermore, we also consider that heterogeneous text data might have different characteristics of noise. These heterogeneous text data can be used to improves robustness of the classifiers. According to Wu (1996), two different noise can be distinguished in a specific dataset, namely class noise and attribute noise. Class noise appears when an example is mislabeled due to data entry errors or there is information shortage when performing the labeling. Further, class noise can be differentiated into two types, which are contradictory examples and misclassification. Contradictory examples represent that duplicate examples have different class labels (Hernández and Stolfo, 1998), while misclassification represents the mislabeled examples that are different from the actual class label (Zhu et al., 2003). Meanwhile, attribute noise bring the meaning that corruptions in the attribute values of instances in a dataset, such as incomplete attributes, missing or unknown attribute values, and erroneous attribute values. In this paper, we focus on class noise because the text data matches with the two types of class noise. In other words, there might be duplicate text examples which have different class labels and possibility of misclassified.

Robustness is important when cope with noisy data, because it is the ability of an algorithm to construct classifiers which are not sensitive to data corruptions and less influenced by the noise in a dataset. In other words, the more robust an algorithm is, the more similar the models built from clean and noisy data are. Therefore, many attempts have been made to enhance the noise-robustness of the classifiers by manipulating with noisy data (Agarwal et al., 2007; Kim et al., 2011; Sáez et al., 2013).

2.2. Semi-Supervised Learning

In machine learning fields, there are two traditional approaches; Supervised learning and unsupervised learning. In order to construct the classification model, a set of sufficient labeled examples are needed or performing the traditional supervised learning algorithms (Witten et al., 2016). The learning results will then use to predict the labels of the unlabeled examples. Different with supervised learning, unsupervised learning is mostly based on unlabeled examples. In details, unsupervised learning does not need any labeled examples to train a model and more into discovering the hidden pattern of unlabeled examples. There has been tremendous amount of online data spread through the web. Those online data can be mostly founded on the news, blogs, online

forums, social networks and it keep increasing constantly. However, most of the accessible online data does not have their own labels which actually hard to apply these data in many real-world practical fields, such as speech recognition, text classification, and sentiment analysis. Moreover, labeled examples are usually time consuming, expensive, and difficult to obtain, because experienced human efforts are needed for the labeling. The important thing is, learning a classifier by using only a little amount of labeled training data may not guarantee satisfied performance. Furthermore, with the condition where labeled data is insufficient, researchers from different fields have attempted to suggest or modify the algorithms by utilizing and manipulating unlabeled data in the learning process for more excellent classification result. In 1968, it was suggested that labeled and unlabeled data could be combined to build classifiers with the likelihood of maximization by testing all possible class assignments (Hartley and Rao, 1968). Additionally, in the field of machine learning, the combined use of labeled and unlabeled examples have been found effective for different tasks (Seeger, 2000). Therefore, SSL was introduced, as this method uses not only the labeled examples but also unlabeled examples to construct a classifier (Zhu and Goldberg, 2009). The goal of semi-supervised learning is to utilize unlabeled examples to build better classifiers with higher accuracy when there are less labeled training examples (Chapelle et al., 2006a).

There are several different methods for SSL which can be roughly categorized into Expectation Maximization (EM) based methods (Nigam et al., 2000), self-training (Li and Zhou, 2005; Rosenberg et al., 2005; Tanha et al., 2017; Yarowsky, 1995), co-training (Blum and Mitchell, 1998; Tanha et al., 2011), Transductive Support Vector Machine (TSVM) (Joachims, 1999), Semi-Supervised SVM (S3VM) (Bennett and Demiriz, 1999), graph-based methods (Zhu et al., 2005), and boosting based SSL methods (Mallapragada et al., 2009).

Other than that, self-labeled techniques are also well-known approaches in SSL. These techniques tend to expand the labeled dataset(s), according to their most confident predictions in classifying the unlabeled examples. Based on Zhu and Goldberg (2009), there are two types of self-labeled techniques: self-training and co-training. For self-training, it is a simple SSL technique that being practically applied in various fields (Ando and Zhang, 2005; Li and Zhou, 2005; Maulik and Chakraborty, 2011). In the process of self-training, a classifier is first learned with a limited amount of labeled examples and the learned classifier will be used to classify the unlabeled examples. The most confident predictions of unlabeled examples will be selected and retrained again by expanding its labeled training set. The self-training classifier model accepts that the predictions by its own are aim to be correct and does not assign any specific assumptions for the input data. (Triguero et al., 2015). Although the algorithm is simple, but it is hard to decide the convergence of it. Generally, the labeling and retraining process are repeated for a certain maximum iteration number of times or achieving until some heuristic convergence standard, which it might be no unlabeled examples left for the iteration. The self-training algorithm performance deeply relies on the labeled data newly selected at each iteration of the training phase. Furthermore, the selection scheme is depending on the prediction confidences. Therefore, it is important to correctly measure the prediction confidence or other words probability estimation in the self-training process. Self-training is the simplest and adaptable approach which was used as the foundation in this study in order to utilize the heterogeneous data in the classi-

fication learning process.

In addition, various natural language processing tasks are conducted by using self-training approach. Yarowsky (1995) uses self-training for word sense disambiguation. A self-training algorithm is used to identify subjective nouns in Riloff et al. (2005) work. A semi-supervised self-training approach using a hybrid of Naive Bayes and decision trees is used to classify sentences as subjective or objective (Wang et al., 2007). However, Manning et al. (2014) stated that the labeled training data is always rare to be obtained, so the initial inductive models are usually hard to provide high confidence predictions. Thus, the accuracies of candidates' labels cannot be guaranteed. Based on previous studies, the performances of semi-supervised learning approaches are not always outperform, sometimes it may be even worse than other approaches that only uses labeled data (Cozman et al., 2003; Grandvalet and Bengio, 2005; Nigam et al., 2000). Moreover, the other studies also show that S3VMs may degrade the performance by using unlabeled data (Chapelle et al., 2006b; Chapelle et al., 2008; Wang et al., 2003). Additionally, since self-training model utilizes unlabeled examples in an incremental manner, the candidates might be consistently mislabeled which will make the model even worse in the next iteration. Thus, resulting in severe performance degradation. In previous studies, even though data editing (Wang et al., 2010) have been employed to alleviate this noise-related problem (Li and Zhou, 2005), the results are undesirable. Therefore, this technique could always suffer from introducing too many wrongly labeled candidates to the labeled training set, which may severely degrade performance.

2.3. Ensemble Learning

Ensemble learning is a machine learning paradigm that has been popularly adopted to combine multiple learners to improve overall prediction/classification accuracy (Dietterich, 2000). In details, compared to machine learning approaches that learn one hypothesis from training data, ensemble learning methods attempt to generate a set of hypotheses and combine them to use (Wang et al., 2011; Zhou, 2012). The earliest works on ensemble learning in Tukey (1977) paper, suggested the combining of two linear regression models. The first linear regression model was fitted to the original data and the second linear model was fitted to the residuals. Two years later, Dasarathy and Sheela (1979) discussed to partition the feature space using two or more classifiers. Furthermore, Hansen and Salamon (1990) proved that the ensemble of similarly configured ANNs can enhanced the generalization performance of a single ANN. At the same time, Schapire (1990) showed that it is possible to generate a strong classifier in probably approximately correct (PAC) sense by combining weak classifiers through Boosting, which is the early version of AdaBoost algorithms. Furthermore, the quality and robustness of unsupervised task also can be improved by using ensemble methods (Dimitriadou et al., 2003). Since then, related studies for ensemble systems have increased rapidly with different creative names and ideas.

The existing approaches are usually differing from each other in terms of the particular step applied for constructing individual classifiers and the scenario applied for combining the classifiers. Generally, two types of combination available, which are classifier selection and classifier fusion (Kuncheva, 2001; Woods et al., 1997). In classifier selection, each classifier was trained to become an expert in certain vicinity

of the total feature space. Based on the given feature vector, the classifiers were combined. According to certain distance metric sense, the classifier trained with data closest to the local area of the feature vector is assigned with the highest confidence. Then, one or more local experts will be designated for the decision making (Alpaydin and Jordan, 1996; Giacinto and Roli, 2001; Jacobs et al., 1991). Nevertheless, in classifier fusion, all the classifiers are trained on the whole feature space. In order to generate a powerful expert classifier with excellent performance, the combination of classifiers is performed by merging the weaker classifiers.

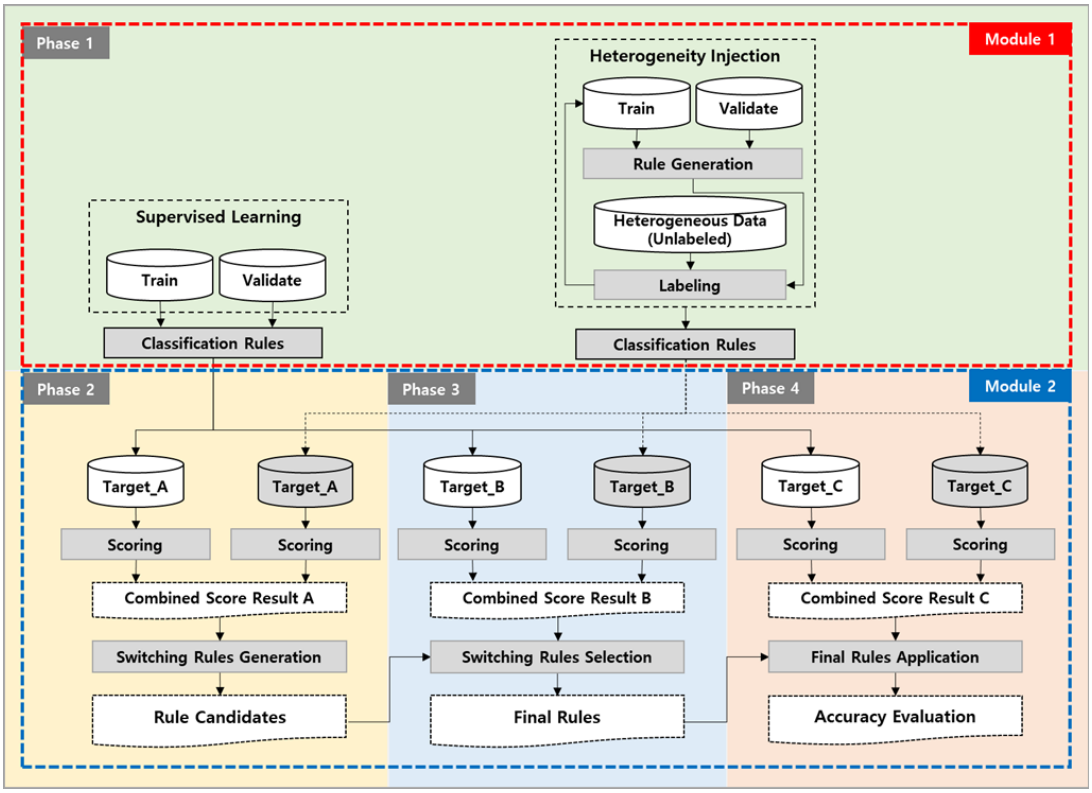
With the existence of noise, outliers and overlapping data distribution, it is impossible to create a classifier with perfect generalization. At best, the classifiers were expected to classify the corresponding data correctly in most of the time. Therefore, generating many classifiers and combining the outputs of those classifiers is the strategy in ensemble systems in order to obtain a combination that is able to improve the performance compared to a single classifier. Nevertheless, the individual classifiers are required to generate errors on different datasets. If different errors were made by each classifier, then the total error can be reducing through a strategic combination of these classifiers. In overall, the main key of ensemble systems is that each classifier has to be as unique as possible, especially to misclassified datasets. In other words, a set of classifiers where their decision boundaries are sufficiently different with each other is consider as diverse (Polikar, 2006). Classifiers diversity can be obtained in several ways, such as (i) using different training datasets to train individual classifiers, (ii) using different training parameters for different classifiers, (iii) using entirely different type of classifiers, and (iv) using different features.

There are two important strategies in constructing an ensemble system. The first one as mention above is to generate an ensemble that is diverse. Some popular ensemble methods were introduced, such as bagging (Breiman, 1996), boosting (Freund and Schapire, 1996; Schapire, 1990), Adaboost (Freund and Schapire, 1997), stacked generalization (Wolpert, 1992), and mixture of experts (Jacobs et al., 1991; Jordan and Xu, 1995). A second strategy describes that the outputs of individual classifiers can be combined into an ensemble model so that the correct decisions can be augmented, and incorrect ones can be negated (Polikar, 2006). The key idea of most methods for building ensemble of classifiers is to modify the training dataset, builds classifiers on these new training sets and then combines them into a final decision rule. In our study, we proposed an ensemble that perform two different classifiers - supervised classifier and heterogeneity classifier on the same training dataset and the outputs will be amplified through a proposed rule selection algorithm into a final decision rule.

Ⅲ. Research Methodology

3.1. Research Overview

This section introduces a methodology to enhance the performance of text classification through the proposed algorithm by using the heterogeneous data. Heterogeneous data refers to unclassified data from other sources without any labels. <Figure 3> shows a research overview, where the cylindrical shapes represent learning data (Train, Validate) and classification target data (Target_A, Target_B, Target_C). Furthermore, the rectangular shapes represent the main processes and the dotted lines represents the



<Figure 3> Research Overview

output of each process.

The proposed methodology consists of two modules: Module 1 heterogeneity injection (Phase 1) and Module 2 classification rule selection (Phase 2 ~ 4). The heterogeneity injection of Module 1 which corresponding to Phase 1 is the core module of this study which performs heterogeneity learning that artificially injecting the heterogeneity into original data in the classification learning process. Heterogeneity learning is performed by adding new features extracted from heterogeneous data to original data. Specifically, an initial classifier is learned by using the original data and the generated rules will be applied to the unlabeled heterogeneous data. The labeled data with the highest predicted value will be appended to the learning data. Since each heterogeneity classifier is generated based on the number of heterogeneous data sources, the existing ensemble learning approach is applied. The prediction results of the heterogeneity classifiers are combined based on ensemble learning approach to select the classification rules that have the highest prediction value. The derived heterogeneity learning-based classification rules and the derived supervised learning-based classification rules will be utilized in Module 2. Classification rule selection of Module 2 in Phase 2 ~ Phase 4 is done by selecting the appropriate classification rules from the derived heterogeneity learning-based classification rules and the derived supervised learning-based classification rules. Based on this process, the ensemble document classifier is build. In this case, the target data is divided into

three data sets (news data) of A, B, and C, in order to select the classification rules for the document classification. Next, the final classification rules will be tested through the evaluation process to verify the performance of the document classifier. Specifically, in Phase 2, the supervised learning-based classification rules and the heterogeneity learning-based classification rules will be applied on the target data A respectively. The scoring will then perform, and both scoring results will be combined. The rules with the highest predicted value will be selected from the combined results of target data A and the selected rules will be used to generate a classification rule candidate group. In other words, according to the characteristics of the target data, the most suitable classification rules will be calculated and to be used in generating the classification rule candidate group for building a document classifier.

Phase 3 is a phase of selecting the classification rules that only contribute to improving the performance of the document classifier among the classification rule candidates derived in Phase 2. The target data B will be scored in the same manner as in Phase 2 by applying the classification rule candidate group derived through the combined results of Phase 2. The classification rules that correctly classify the target data B are selected as the final classification rules. In Phase 4, the final classification rules verified from Phase 3 will be used to construct the document classifier and the prediction accuracy will be evaluated. In other words, the target data C will be scored in the same manner as in Phase 2 and 3. The scored results will be combined and the prediction accuracy of applying the final classification rules on target data C will be evaluated. The further details of the proposed methodology will be explained in section 3.2. and 3.3.

3.2. Module 1: Heterogeneity Injection

Specifically, this section will describe the detail process of heterogeneity learning that extracts new features from different heterogeneous data through self-training approach of semi-supervised learning and the extracted features will be added into the learning data.

3.2.1. Data Structurization

Prior to the implementation of the proposed methodology, the data structurization should be first performed. In the case of text document, it is necessary to convert the unstructured data into a structured analyzable format. In this study, a representative technique of text mining which is topic modeling will be applied for this purpose. Topic modeling grouping similar documents based on the frequency of terms appears in each document, then extracting key terms that representing each group and presenting a set of topic keywords for that corresponding group (Blei et al., 2003; Hofmann, 2001). For example, if some terms appear simultaneously in various documents, the documents can be considered similar and grouped into the same topic. In most cases, topic modeling allows each document to be simultaneously links to multiple topics and describes a topic as a recurring pattern of co-occurring terms.

Topic modeling is performed according to the weighted frequency of terms in a document corpus. Term frequency-inverse document frequency (TF-IDF) are applied to measure the relative importance of terms in each document, because TF-IDF is able to capture the amount of information a term provides (Provost and Fawcett, 2013). TF-IDF is a computation method that assigns low value to common terms that frequently appearing in many docu-

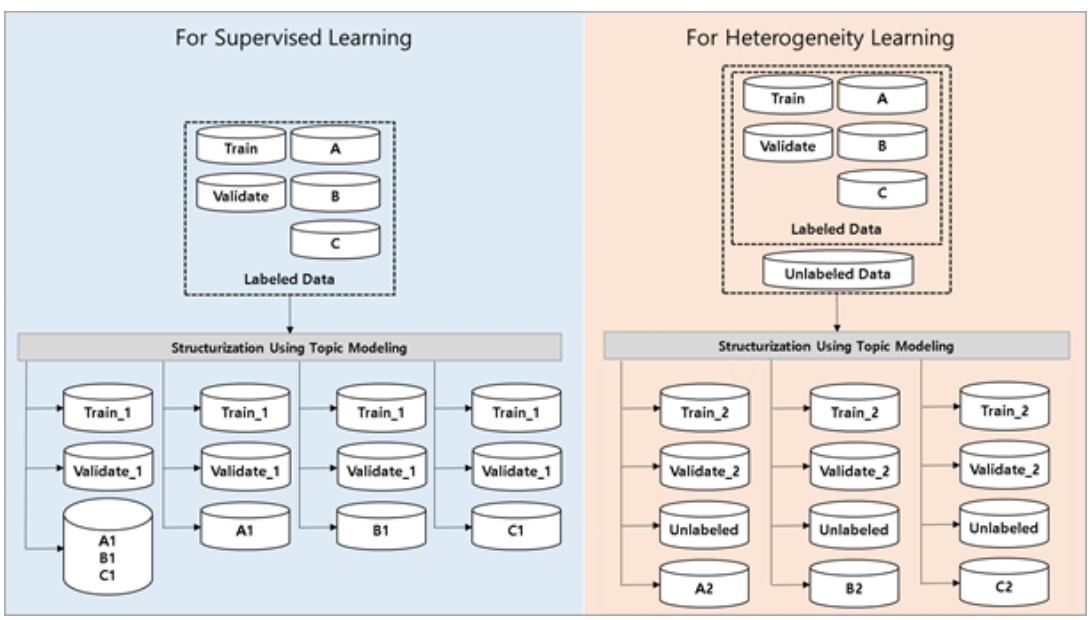
ments and gives $high_d$ value to specific word that appearing only in specific documents. Each document is representing by a vector with the number of terms and their TF-IDF value. TF-IDF value is measured in the following manner.

$$TF(d,t) = \begin{cases} 0 & if\ freq(d,t)=0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise \end{cases}$$

$$IDF(t) = \log\frac{|d|}{|d_t|}$$

In the above equation, freq (d, t) is the term frequency of term t appears in document d, |d| is the total number of documents, and $|d_t|$ represents the number of documents containing the term t. If the term t does not appear in other documents, but only appears in specific documents, then the value of TF-IDF (d, t) is high. Based on this equation, the document topic weight of an individual document toward each topic was calculated, which is the normalized sum of the TF-IDF weightings for each term in the document multiplied by the topic weights. The document topic weights. The document topic weight for each document with the document corresponding label will be used to construct the classifier in the next phase.

<Figure 4> shows the data structurization process for different type of learning. Data structurization is accomplished by performing topic modeling on the integrated experimental data to which each classifier is applied. If there are N number of classifiers will be constructed, then N times of topic modeling will be performed for the data structurization. The data structurization process is performed according to the type of classifier, because the difference of the characteristics extracted from the heterogeneous data might affect the topic weights. For example, suppose that the supervised classifier was constructed only based on news data, while heterogeneity classifier was constructed not only based on news data, but also with additional heterogeneous data such as Twitter and blog data. Therefore, there will be three



<Figure 4> Data Structurization for Learning

<Table 1> Example of Unstructured Text Data

	Label	Content
Doc_1	ENTERTAIN	강부자 과거모습이 공개됐다.강부자는 6월 22일 방송된 SBS '힐링캠프 in
Doc_2	DIGITAL	디스플레이 기기 전문 업체 알파스캔모니터(대표 류영렬)가 4K(3840×21(
Doc_3	ENTERTAIN	인기가요 백현, 열애설 후 첫 스케줄 "표정 변화 없이..."태연과의 열애를
Doc_4	POLITICS	동부전선 GOP에서 총기난사 사건이 발생한 육군 22사단에는 이른바 '관
Doc_5	SPORTS	'2014 브라질 월드컵' 대한민국과 알제리전이 펼쳐지기 전인 22일 저녁,

<Table 2> Example of Data Structurization Result

	Topic_1	Topic_2	Topic_3	Topic_4	Topic_5	Label
Doc_1	0.004	0.000	0.074	-0.004	0.000	ENTERTAIN
Doc_2	-0.004	0.000	0.163	-0.001	0.009	DIGITAL
Doc_3	0.004	-0.001	0.002	-0.005	0.011	ENTERTAIN
Doc_4	0.003	-0.001	0.002	-0.003	-0.003	POLITICS
Doc_5	0.009	0.000	0.060	-0.004	0.012	SPORTS

different classifiers constructed and also means that there will be three times of topic modeling performed. <Table 1> shows the example of unstructured text data before performing the data structurization. After the data structurization was performed, the unstructured text data will be converted into structured analyzable format and the structurization result was shown in <Table 2>.

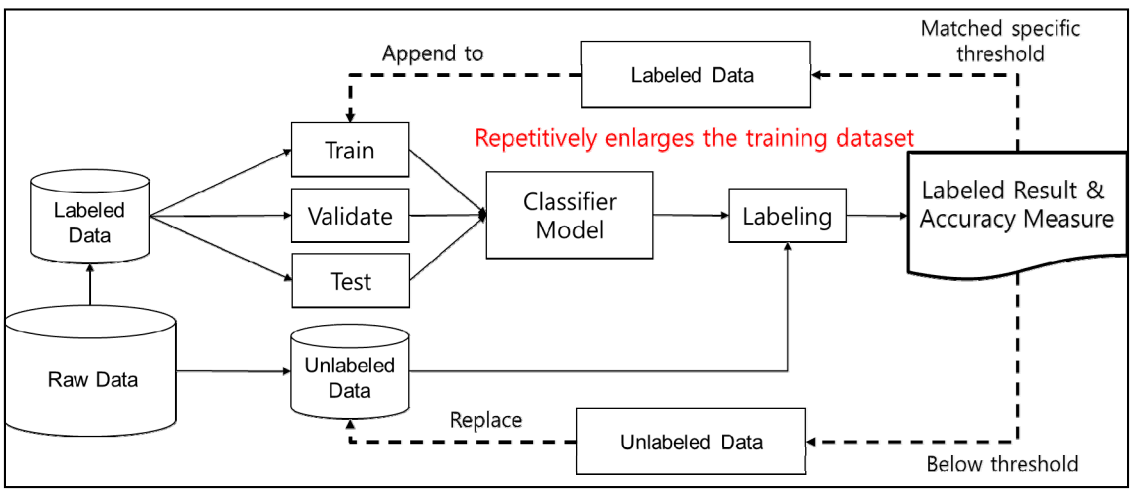
3.2.2. Phase 1: Supervised Learning Classifier

There are two classification models were constructed in Module 1, which are supervised learning classifier and heterogeneity learning classifier. For supervised learning classifier, the structured dataset prepared from the data structurization process will be used to construct the classifier. The learning dataset were divided into train data and validate data. Based on this, the chosen classification model will be applied and the constructed model will be used as classifier. In this study, three supervised learning classification models were first used to apply on the learning dataset and the performance of each model were compared. The three supervised learning models are Decision Tree, Support Vector Machine (SVM), and Neural Network. In order to measure the performance of each model, misclassification rate is applied and calculated. The lower the misclassification rate, the more appropriate the classification model for this study. After comparing the misclassification rate of each model, Neural Network has the lowest misclassification rate than either Decision Tree and SVM. Therefore, Neural Network is chosen as the main classification model for this study. Based on the selected supervised learning model, the supervised learning-based classification rules will be derived through performing the learning on the original labeled data. Further, the derived classification rules will be used to score the target data in Module 2.

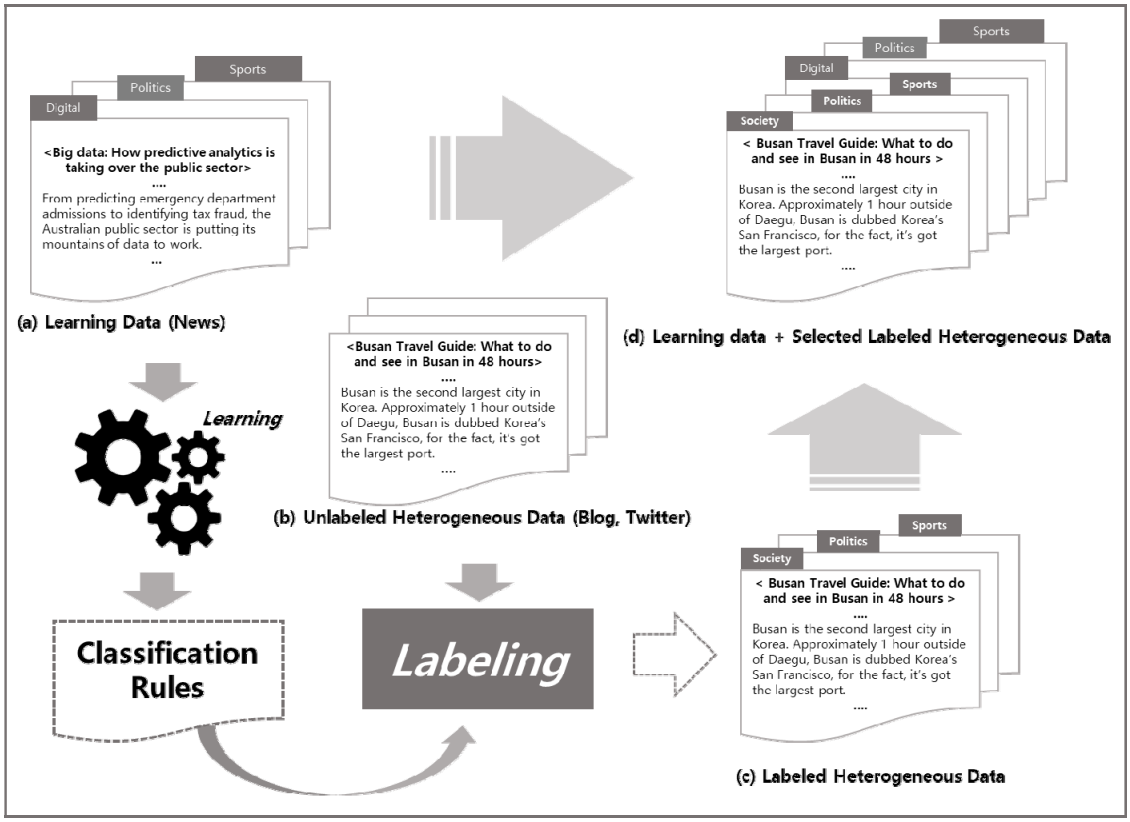
3.2.3. Phase 1: Heterogeneity Learning Classifier

The heterogeneity learning classifier is constructed through the existing machine learning classifier by applying the self-training approach on the learning data. <Figure 5> shows the process of constructing the traditional self-training classifier. The raw data consists of two different set of data, which are labeled data and unlabeled data. After structurization of the raw data, the structured labeled data were further divided into train, validate, and test data. Based on these data, the classifier model will be constructed. The generated classification rules will be used to label the structured unlabeled data. According to certain confidence threshold, the unlabeled data with high confidence prediction and their predicted labels are sorted out. The new sorted data will then append to the training dataset. By using the new learning dataset, the classifier is re-trained and the classification process is repeated. Next, the new generated classification rules will be used to label the remaining unlabeled data. According to this approach, most of the unlabeled data can be converted into labeled data and this approach can repetitively enlarge the training dataset. Noted that the labeled data and unlabeled data both are from the same data source, which is homogeneous data.

<Figure 6> shows an example of heterogeneity injection process through heterogeneity learning. In <Figure 6>, (a) the news data, which also is the original data were used for learning in order to generate the initial classification rules. Different with traditional self-training approach, the initial classification rules were used to label the unlabeled heterogeneous data instead of unlabeled homogeneous data. As shown in <Figure 6>, (b) the classification rules were applied to the unlabeled blog and Twitter data which are different with the news data. Then, the labeled heterogeneous data with high confidence prediction, along with their predicted labels are selected and appended to the learning dataset. By using the new appended learning dataset, the classifier is re-trained and the classification rules are re-generated. The final classification rules are selected through iterative learning process and used to construct the document classifier. By utilizing the unlabeled heterogeneous data for learning, the heterogeneity classifier can be constructed through



<Figure 5> Self-training Approach



<Figure 6> Heterogeneity Injection Process

modifying the learning data, which is injecting the heterogeneity feature into the original learning data and indirectly enriching the feature dimension of the learning data. Therefore, several heterogeneity classifiers can be generated according to the number of heterogeneous data sources. Furthermore, the final heterogeneity learning-based classification rules can be derived by selecting the classification rules having the highest prediction value through the combined prediction results of the heterogeneity classifiers. Similar with supervised learning-based classification rules, the heterogeneity learning-based classification rules will be used to score the target data in Module 2.

3.3. Module 2: Classification Rule Selection

This section corresponds to Phase 2 ~ Phase 4 of <Figure 3>. Based on the supervised learning-based classification rules and heterogeneity learning-based classification rules derived in Module 1, this section will introduce the process of constructing the final document classifier using an ensemble learning algorithm.

3.3.1. Phase 2: Switching Rules Generation

This subsection is corresponding to Phase 2 of Module 2, where the derived classification rules of both supervised classifier and heterogeneity classifier

will be used to apply on the target data. Based on <Figure 3>, the classification rules derived by supervised classifier and heterogeneity classifier will be applied on target data A, B, and C respectively in Module 2. Further, the scored results of target A by both supervised classifier and heterogeneity classifier will be combined in Phase 2. In details, the confidence value and predicted labels (category) for each documents in both target data will be combined and the example of the combined scored result is shown in <Table 3>. In <Table 3>, SC represents supervised classifier and HC represents heterogeneity classifier.

Based on the example above, the difference of confidence value for each document in both target data will be calculated. First of all, the confidence value and predicted label of each document derived by supervised classifier will be used as default. If the difference value between supervised classifier and heterogeneity classifier is positive, then the predicted label derived by heterogeneity classifier will be used to replace the default label. For example, as the difference value of document 1001 and 1003 are shown positive in <Table 3>, the default label "SPORTS" of document 1001 is replaced by "DIGITAL" and the default label "DIGITAL" of document 1003 is replaced by "POLITICS." The corresponding decision result is shown in <Table 4>.

However, it will be too risky if replace the default label by using the above method. In the case of document 1003 in <Table 4>, the actual label (Ori_Category) is "DIGITAL", which may be misclassified as "POLITICS" according to the above switching rule. If the document is classified by using the switching rule generated based on the difference of the confidence value, the performance may be deteriorated due to the incorrect switching. Therefore, it is necessary to calculate the correctness of the label replacement by comparing the replaced label with the original label of target data. In order to generate the appropriate rules, a ensemble-based rule selection algorithm (ERSA) was proposed by using (1) different threshold of positive difference value and (2) correctness of the replaced label. The ensemble rule selection algorithm is performed as follows.

<Table 3> Example of Combined Scored Result for Target Data A

Doc_No	SC_Confidence	SC_Category	HC_Confidence	HC_Category	Difference
1001	0.404	SPORTS	0.863	DIGITAL	0.458
1002	0.844	ENTERTAIN	0.514	DIGITAL	-0.330
1003	0.682	DIGITAL	0.977	POLITICS	0.296
1004	0.989	DIGITAL	0.682	DIGITAL	-0.307

<Table 4> Example of Combined Scored Result with Original Category

Doc_No	Ori_Category	SC_Confidence	SC_Category	HC_Confidence	HC_Category	Difference	Decision
1001	DIGITAL	0.404	SPORTS	0.863	DIGITAL	0.458	DIGITAL
1002	DIGITAL	0.844	ENTERTAIN	0.514	DIGITAL	-0.330	ENTERTAIN
1003	DIGITAL	0.682	DIGITAL	0.977	POLITICS	0.296	POLITICS
1004	DIGITAL	0.989	DIGITAL	0.682	DIGITAL	-0.307	DIGITAL

```
Loop SC & HC Confidence Result Rows
If Diff < 0 OR HC_Cat == SC_Cat OR (ORG_Cat != HC_Cat and ORG_Cat != SC_Cat) then Continue
Endif
If ORG_Cat == HC_Cat and ORG_Cat != SC_Cat then CorrectCount = CorrectCount + 1
Elseif ORG_Cat <> HC_Cat and ORG_Cat == SC_Cat then IncorrectCount = IncorrectCount + 1
Endif
CurrentDiffGain = CorrectCount - IncorrectCount
If (Current Row Rule == Next Row Rule) AND (Current Row Diff == Next Row Diff) then Continue
Elseif(Current Row Rule != Next Row Rule) then
    NetGain = CurrentDiffGain
Elseif(Current Row Rule == Next Row Rule) AND
    (Current Row Diff != Next Row Diff) then
    If NetGain < CurrentDiffGain then
    NetGain = CurrentDiffGain
    Endif
Endif
If NetGain > 0
    Print Selected Rule
    NetGain = 0
    CorrectCount = 0
    IncorrectCount = 0
Endif
End Loop2
```

By using the proposed algorithm, the switching rule candidates were generated according to different thresholds. <Table 5> shows an example of the switching rule candidates generated by the proposed algorithm. The net gain is calculated according to the number of correct switched and incorrect switched by comparing the predicted label with the original label. The switching rule candidates with net gain more than zero were selected and will be used in the Phase 3. Based on <Table 5>, the switching rule candidates R1, R2, R5, R6, and R7 were derived and will be applied to target data B in Phase 3.

3.3.2. Phase 3: Switching Rules Selection

Phase 3 of Module 2 is the phase where the generated switching rule candidates will be verified and the final switching rules will be selected. As mention before, the switching rules derived by both different classifiers in Phase 1 were applied to target data B in Phase 3. The scored results of target B by both supervised classifier and heterogeneity classifier will be combined. Next, the generated rule candidates R1, R2, R5, R6, and R7 derived in Phase will be used to apply on the combined score result B. Same as Phase 2, the score result of supervised classifier is set as default. According to the threshold of each rule candidate, the default label of each document in target dataset B will be replaced by heterogeneity classifier label. After applying the rule candidates, the correct switched count and incorrect switched count of each rule candidate will be calculated and the net gain is obtained. Based on the net gain of each switching rule candidate, the rule candidates with net gain smaller than one will be disqualified. By using rule candidate R7 as example, the “Sports” label documents with the difference value (Threshold) more than 0.4 were replaced with the label “Digital”. However, as the net gain in target data B is smaller than one, the rule candidate R7 is considered invalid and will be disqualified from the final switching rule list. The example of qualified switching rule candidates was shown in <Table 6>.

3.3.3. Phase 4: Final Rules Application

After the validation process in Phase 3, the final switching rules R1, R2, R5, and R6 were selected

<Table 5> Example of Switching Rules Generation

Rule No	Rule	Threshold	Correct	Incorrect	Net Gain	Selected Gain (>0)
R1	DIGITAL->ENTERTAIN	0	12	0	12	TRUE
R2	DIGITAL->POLITICS	0	42	14	28	TRUE
R3	ENTERTAIN->DIGITAL	0	21	38	-17	FALSE
R4	ENTERTAIN->POLITICS	0.1	8	9	-1	FALSE
R5	ENTERTAIN->POLITICS	0	12	5	7	TRUE
R6	ENTERTAIN->SPORTS	0	39	20	19	TRUE
R7	SPORTS->DIGITAL	0.4	1	0	1	TRUE
R8	SPORTS->DIGITAL	0.3	1	1	0	FALSE
R9	SPORTS->DIGITAL	0.2	1	3	-2	FALSE
R10	SPORTS->DIGITAL	0.1	2	4	-2	FALSE
R11	SPORTS->DIGITAL	0	7	7	0	FALSE
R12	SPORTS->ENTERTAIN	0	13	13	0	FALSE

<Table 6> Example of Rule Candidates After Switching Rules Selection

Rule No	Rule	Threshold	Correct	Incorrect	Net Gain	Selected Gain (>0)
R1	DIGITAL->ENTERTAIN	0	11	6	5	TRUE
R2	DIGITAL->POLITICS	0	30	12	18	TRUE
R5	ENTERTAIN->POLITICS	0	15	3	12	TRUE
R6	ENTERTAIN->SPORTS	0	43	18	25	TRUE
R7	SPORTS->DIGITAL	0.4	3	5	-2	FALSE

and will be applied to the target data C without any modification in Phase 4. According to the threshold of each final switching rules, the default label of each document in target dataset C will be replaced by heterogeneity classifier label. Lastly, the replaced labels of the documents in target dataset C was compared with the original labels and the prediction accuracy of the proposed methodology on target dataset C was calculated. In order to evaluate the performance of the proposed methodology, the whole process of classification rule selection (Module 2) will be repeated for twice until the prediction accuracy of the proposed methodology on target dataset A and B were obtained.

Ⅳ. Data Analysis and Results

4.1. Data Description

There are three types of data sources were used, which are news, Twitter, and blog data. Total 387,018 news articles, 327,554 blog articles, and 14 million tweets between 22 June 2014 and 5 July 2014 were collected. The news articles are collected from South Korea news portal Daum news, while the blog articles are collected from South Korea weblog posts in Naver blog. Among the collected news articles, the news articles were further extracted based on their categories/labels, which are Digital, Entertainment, Politics, and Sports. However, the “Digital” category was

found to have the smallest number of articles, which is 8,250 articles. In order to maintain the equity between the categories, the number of news articles for each category are extracted based on the number of news articles available in "Digital" category. Therefore, a total of 33,000 news articles were extracted with 8,250 articles for each category. Follow by that, the preprocessing of the collected data was performed based on the stop words dictionary in order to remove the stop words and punctuation. Further, the preprocessed news articles are divided into learning dataset, target dataset and unlabeled news dataset. By random sampling, there are 1,000 news articles were extracted as labeled learning dataset and 20,000 news articles as unlabeled news dataset. In order to evaluate the accuracy of the proposed methodology, there are 12,000 news articles (3,000 news articles for each category) were selected as target dataset. The target dataset will then further divide into three target datasets A, B, and C (4,000 news articles for each dataset). Furthermore, 20,000 blog articles and 200,000 tweets were randomly selected from those collected data. The blog articles will be used as unlabeled blog dataset and the tweets will be used as unlabeled twitter dataset.

4.2. Data Preparation

First of all, all the experimental data need to go through the data structurization process in order to convert the unstructured text data into structured data. According to <Figure 4>, the experimental data were first reorganized into different datasets according to their corresponding purpose. The dataset used for constructing the supervised learning classifier consists of learning dataset and target dataset, while the dataset used for constructing the heterogeneity learning classifier consists of learning dataset, target dataset, and unlabeled dataset. As there are three different unlabeled datasets, which are news, twitter, and blog datasets. Therefore, there are four datasets were used for constructing the classifiers, which are supervised dataset, homogeneity news dataset, heterogeneity twitter dataset, and heterogeneity blog dataset. These datasets will then proceed to data structurization process.

After the structurization of the datasets, the 1,000 labeled learning data were further divided into 600 train data and 400 validate data according to the ratio 6:4. Furthermore, 20,000 unlabeled news data, 20,000 blog data, and 200,000 Twitter data were separated and used as unlabeled data according to their corresponding purpose. Moreover, 12,000 labeled news data were divided into target dataset A, B, and C with each dataset contains 4,000 labeled news data (1,000 data for each category).

4.3. Experiments and Results

4.3.1. Selection and Combination of Scored Results

In the process of constructing the initial classifier, the Artificial Neural Network (ANN) model was chosen to be the main classification model based on the performance comparison results of each model. In the case of supervised classifier, the classification rules are generated through ANN model by using supervised dataset. The generated classification rules will be directly used to score the target dataset A, B, and C. On the other hand, for semi-supervised classifier, the generated rules will be first used to label the unlabeled dataset. Three different unlabeled datasets (news, blog and twitter dataset) will be used accordingly. The injection process was performed according to <Figure 6>. As a result, with

<Table 7> Scored Results Selection (Part)

No	News		Blog		Twitter		ERSA	
	Confidence	Category	Confidence	Category	Confidence	Category	Confidence	Category
1028	0.973	DIGITAL	0.422	SPORTS	0.795	DIGITAL	0.973	DIGITAL
1305	0.976	DIGITAL	0.995	DIGITAL	0.876	DIGITAL	0.995	DIGITAL
1337	0.977	DIGITAL	0.985	DIGITAL	0.696	DIGITAL	0.985	DIGITAL
2047	0.953	ENTERTAIN	0.880	SPORTS	0.994	SPORTS	0.994	SPORTS
2050	1.000	ENTERTAIN	0.864	DIGITAL	0.621	ENTERTAIN	1.000	ENTERTAIN
2059	0.695	SPORTS	0.749	SPORTS	0.994	SPORTS	0.994	SPORTS
3085	0.977	POLITICS	0.816	POLITICS	0.967	POLITICS	0.977	POLITICS
3274	0.971	POLITICS	0.623	SPORTS	0.470	SPORTS	0.971	POLITICS
3276	0.980	POLITICS	0.951	POLITICS	0.992	POLITICS	0.992	POLITICS
4931	0.972	DIGITAL	1.000	ENTERTAIN	0.389	SPORTS	1.000	ENTERTAIN
4762	0.874	SPORTS	0.691	SPORTS	0.994	SPORTS	0.994	SPORTS
4928	0.991	SPORTS	0.933	SPORTS	0.994	SPORTS	0.994	SPORTS

the threshold of more than 0.9, there are newly labeled 18, 263 news data, 16, 737 blog data, and 101,100 twitter data were selected and appended to their initial learning set respectively. The initial classifier was re-trained, and the new generated classification rules were used to score the target dataset A, B, and C. In order to derive the best classification rules, the predicted confidence values and the predicted labels of each document for three different classifiers were combined. The highest predicted value and the corresponding label of each document were selected through the combined scored result shown in <Table 7>. The derived result will be used as ERSA classifier.

The scored result of the supervised classifier and ERSA classifier were combined and the difference values between the predicted confidence values of both classifiers for each document were calculated. The derived combined scored result was shown in <Table 8>. The original label of each document is also included in <Table 8>.

4.3.2. Classification Rule Selection

Based on the combined scored result A, the switching rule candidates were derived using the proposed ERSA in Phase 2. The generated switching rule candidates were shown in Phase 2 of <Table 9>. The switching rule candidates were verified by applying those rule candidates to target dataset B in Phase 3. After the verification process in Phase 3, the switching rule candidates with net gain smaller than one will be disqualified as shown in Phase 3 of <Table 9>. Finally, in Phase 4, the qualified switching rules were applied on target dataset C without any modification. The classification accuracy of the ERSA classifier for target dataset C was calculated. The same process of Phase 2 ~ 4 repeated twice for the following sequences: BCA and CAB in order to obtain the classification accuracy of target dataset A and B. The process for BCA and CAB in each phase were shown in <Table 9>.

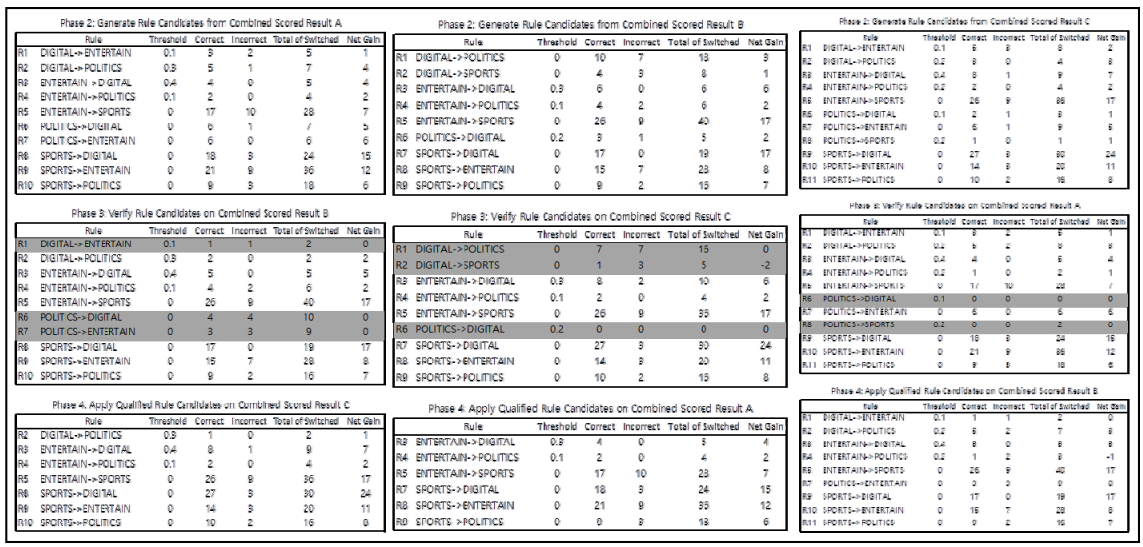
4.3.3. Methodology Evaluation

As a result, the classification accuracy of the ERSA classifier for target dataset A, B, and C were calculated and the total result was derived. In addition, the classification accuracy of the supervised classifier was

<Table 8> Combined Scored Result A with Difference Value (Part)

No	Ori_Category	Supervised Classifier		ERSA Classifier		Difference
		Confidence	Category	Confidence	Category	
1028	DIGITAL	0.908	SPORTS	0.973	DIGITAL	0.065
1305	DIGITAL	0.948	POLITICS	0.995	DIGITAL	0.047
1337	DIGITAL	0.942	POLITICS	0.985	DIGITAL	0.043
2047	ENTERTAIN	0.976	ENTERTAIN	0.994	SPORTS	0.019
2050	ENTERTAIN	0.849	DIGITAL	1.000	ENTERTAIN	0.151
2059	ENTERTAIN	0.998	SPORTS	0.994	SPORTS	-0.004
3085	POLITICS	0.955	POLITICS	0.977	POLITICS	0.022
3274	POLITICS	0.401	SPORTS	0.970	POLITICS	0.569
3276	POLITICS	0.794	ENTERTAIN	0.992	POLITICS	0.198
4931	SPORTS	0.642	SPORTS	1.000	ENTERTAIN	0.358
4762	SPORTS	0.946	ENTERTAIN	0.994	SPORTS	0.048
4928	SPORTS	0.836	ENTERTAIN	0.994	SPORTS	0.158

<Table 9> Switching Rule Candidates of Phase 2 ~ 4



Phase 2: Generate Rule Candidates from Combined Scored Result A

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R1	DIGITAL->ENTERTAIN	0.1	3	2	5	1
R2	DIGITAL->POLITICS	0.3	5	1	7	4
R3	ENTERTAIN->DIGITAL	0.4	4	0	5	4
R4	ENTERTAIN->POLITICS	0.1	2	0	4	2
R5	ENTERTAIN->SPORTS	0	17	10	28	7
R6	POLITICS->DIGITAL	0	6	1	7	5
R7	POLITICS->ENTERTAIN	0	6	0	6	6
R8	SPORTS->DIGITAL	0	18	3	24	15
R9	SPORTS->ENTERTAIN	0	21	9	36	12
R10	SPORTS->POLITICS	0	9	3	18	6

Phase 2: Generate Rule Candidates from Combined Scored Result B

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R1	DIGITAL->POLITICS	0	10	7	18	3
R2	DIGITAL->SPORTS	0	4	3	8	1
R3	ENTERTAIN->DIGITAL	0.3	6	0	6	6
R4	ENTERTAIN->POLITICS	0.1	4	2	6	2
R5	ENTERTAIN->SPORTS	0	26	9	40	17
R6	POLITICS->DIGITAL	0.2	3	1	5	2
R7	SPORTS->DIGITAL	0	17	0	19	17
R8	SPORTS->ENTERTAIN	0	15	7	28	8
R9	SPORTS->POLITICS	0	9	2	15	7

Phase 2: Generate Rule Candidates from Combined Scored Result C

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R1	DIGITAL->ENTERTAIN	0.1	5	3	9	2
R2	DIGITAL->POLITICS	0.2	6	0	6	6
R3	ENTERTAIN->DIGITAL	0.4	8	1	9	7
R4	ENTERTAIN->POLITICS	0.2	2	0	4	2
R5	ENTERTAIN->SPORTS	0	26	9	86	17
R6	POLITICS->DIGITAL	0.1	2	1	3	1
R7	POLITICS->ENTERTAIN	0	6	1	9	5
R8	POLITICS->SPORTS	0.2	1	0	1	1
R9	SPORTS->DIGITAL	0	27	3	80	24
R10	SPORTS->ENTERTAIN	0	14	3	20	11
R11	SPORTS->POLITICS	0	10	2	16	8

Phase 3: Verify Rule Candidates on Combined Scored Result B

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R1	DIGITAL->ENTERTAIN	0.1	1	1	2	0
R2	DIGITAL->POLITICS	0.3	2	0	2	2
R3	ENTERTAIN->DIGITAL	0.4	5	0	5	5
R4	ENTERTAIN->POLITICS	0.1	4	2	6	2
R5	ENTERTAIN->SPORTS	0	26	9	40	17
R6	POLITICS->DIGITAL	0	4	4	10	0
R7	POLITICS->ENTERTAIN	0	3	3	9	0
R8	SPORTS->DIGITAL	0	17	0	19	17
R9	SPORTS->ENTERTAIN	0	15	7	28	8
R10	SPORTS->POLITICS	0	9	2	16	7

Phase 3: Verify Rule Candidates on Combined Scored Result C

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R1	DIGITAL->POLITICS	0	7	7	15	0
R2	DIGITAL->SPORTS	0	1	3	5	-2
R3	ENTERTAIN->DIGITAL	0.3	8	2	10	6
R4	ENTERTAIN->POLITICS	0.1	2	0	4	2
R5	ENTERTAIN->SPORTS	0	26	9	86	17
R6	POLITICS->DIGITAL	0.2	0	0	0	0
R7	SPORTS->DIGITAL	0	27	3	80	24
R8	SPORTS->ENTERTAIN	0	14	3	20	11
R9	SPORTS->POLITICS	0	9	2	15	8

Phase 4: Apply Qualified Rule Candidates on Combined Scored Result C

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R2	DIGITAL->POLITICS	0.3	1	0	2	1
R3	ENTERTAIN->DIGITAL	0.4	8	1	9	7
R4	ENTERTAIN->POLITICS	0.1	2	0	4	2
R5	ENTERTAIN->SPORTS	0	26	9	86	17
R8	SPORTS->DIGITAL	0	27	3	80	24
R9	SPORTS->ENTERTAIN	0	14	3	20	11
R10	SPORTS->POLITICS	0	10	2	16	9

Phase 4: Apply Qualified Rule Candidates on Combined Scored Result A

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R3	ENTERTAIN->DIGITAL	0.3	4	0	5	4
R4	ENTERTAIN->POLITICS	0.1	2	0	4	2
R5	ENTERTAIN->SPORTS	0	17	10	28	7
R7	SPORTS->DIGITAL	0	18	3	24	15
R8	SPORTS->ENTERTAIN	0	21	9	36	12
R9	SPORTS->POLITICS	0	9	3	18	6

Phase 4: Apply Qualified Rule Candidates on Combined Scored Result B

	Rule	Threshold	Correct	Incorrect	Total of Switched	Net Gain
R1	DIGITAL->ENTERTAIN	0.1	1	1	2	0
R2	DIGITAL->POLITICS	0.2	6	2	7	3
R3	ENTERTAIN->DIGITAL	0.4	8	0	8	8
R4	ENTERTAIN->POLITICS	0.2	1	2	3	-1
R5	ENTERTAIN->SPORTS	0	26	9	40	17
R7	POLITICS->ENTERTAIN	0	3	3	9	0
R8	SPORTS->DIGITAL	0	17	0	19	17
R10	SPORTS->ENTERTAIN	0	16	7	28	8
R11	SPORTS->POLITICS	0	9	2	16	7

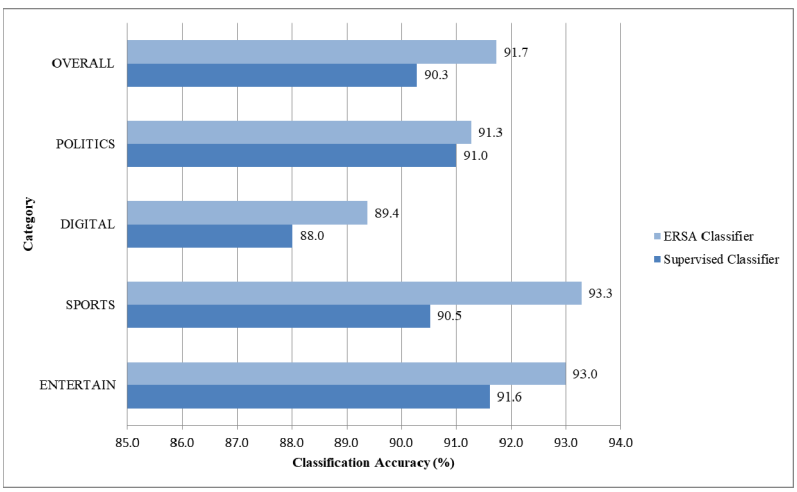
also derived for the comparison. <Table 10> compares the prediction results between the supervised classifier and ERSA classifier. In <Table 10>, by comparing to the actual labels of 12,000 target documents, there are 10,836 documents correctly predicted by supervised classifier and 11,008 documents by ERSA classifier. Based on the predicted and correct labelled results of both classifier, the measurements such as precision, recall, and F1 score were applied and the results were shown in <Table 11>. F1 Score was used to measure the classification accuracy, and percentage point (%p) was used to measure the arithmetic difference of two percentages. Through <Table 11> and <Figure 7>, the classification accuracy be-

<Table 10> Prediction Result Comparison between Supervised and ERSA Classifier

		Supervised Classifier		ERSA Classifier	
Category	Actual	Predicted	Correct	Predicted	Correct
ENTERTAIN	3000	2947	2724	2908	2747
SPORTS	3000	3169	2792	3066	2829
DIGITAL	3000	2831	2566	2912	2642
POLITICS	3000	3053	2754	3114	2790
Total	12000	12000	10836	12000	11008

<Table 11> Performance Comparison between Supervised and ERSA Classifier

	Supervised Classifier			ERSA Classifier		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
ENTERTAIN	92.4	90.8	91.6	94.5	91.6	93.0
SPORTS	88.1	93.1	90.5	92.3	94.3	93.3
DIGITAL	90.6	85.5	88.0	90.7	88.1	89.4
POLITICS	90.2	91.8	91.0	89.6	93.0	91.3
OVERALL	90.3	90.3	90.3	91.8	91.7	91.7



<Figure 7> Classification Accuracy Comparison for Each Category

tween the supervised classifier and ERSA classifier were compared for each category. The result shows that the ERSA classifier was outperformed than the traditional supervised classifier for each category: entertainment (1.4%p), sports (2.8%p), digital (1.4%p), politics (0.3%p). In overall, compared to supervised classifier, the classification accuracy of ERSA classifier improved from 90.3% to 91.7% with total accuracy improvement of 1.4%p.

Ⅴ. Conclusion

In this study, a new concept of heterogeneity learning was proposed by applying the heterogeneous text documents in a self-training process. Furthermore, this study proposed a Ensemble-Based Rule Selection Algorithm (ERSA) as a new method that able to be used to improve the text classification accuracy. The proposed methodology injects heterogeneity into the learning process, and the ERSA was used to manipulating the possible features generated by different classifiers derived from different data sources. Further, among the derived classification rules, only the classification rules that contribute to improving the performance of the document classifier were selected. Aside from news data, heterogeneous source data were collected in two different domains: blog and Twitter data. Based on the results of classification analysis, we observed that our proposed method was outperformed than the traditional supervised classification method with the accuracy improvement of 1.4%p.

The proposed methodology contributes to the academic and practical fields in the following aspects. From the academic viewpoint, the proposed methodology has a significant contribution in that it proposes a method to improve the performance of document classification by performing heterogeneity learning through the utilization of heterogeneous data that having different possible features. This can be regarded as a new attempt in utilizing not only the documents from the same data sources but also the heterogeneous documents from the data sources having different features. This method able to enhance the learning data in the process of constructing the document classifier. In addition, another contribution is that the classification can be performed in a more accurate way by applying the appropriate classification rules according to the features of the specific data through the selection of classification rules from different classifiers. On the practical side, the proposed methodology able to enhance the classification rules derived using the homogeneous data through the utilization of heterogeneous data, therefore contributing to the efficient classification and management of the large amounts of text data that generated in real time.

There are three limitations that need to be overcome in the future. First, the experiment results may vary depending on the characteristic of the unlabeled heterogeneous data, thus the experiment must be repeated by using other potential heterogeneous source data. Second, although the experiments were conducted using only 1,000 learning data and the accuracy of the classification using the heterogeneity learning was improved, but the improvement range is relatively small. Therefore, in future studies, it is necessary to understand how the heterogeneous data will affect the classification accuracy, because the accuracy may be varied according to the amount of heterogeneous data injected in the learning process. In order to enhance the applicability of the proposed method, it is important to automate the analysis processes that being performed manually.

<References>

[1] Agarwal, S., Godbole, S., Punjani, D., and Roy, S. (2007). How much noise is too much: A study in automatic text classification, In *Seventh IEEE International Conference on Data Mining*, 3-12.

[2] Alpaydin, E., and Jordan, M. I. (1996). Local linear perceptrons for classification. *IEEE Transactions on Neural Networks, 7*(3), 788-794.

[3] Ando, R. K., and Zhang, T. (2005). A framework

for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research, 6*(Nov), 1817-1853.

[4] Angelova, R., and Weikum, G. (2006). Graph-based text classification: Learn from your neighbors, In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 485-492.

[5] Aslam, S. (2018). *Twitter by the numbers: Stats, demographics and fun facts*. [Online]. Retrieved from https://www.omnicoreagency.com/twitter-statistics/

[6] Bennett, K. P., and Demiriz, A. (1999). Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems*, 368-374.

[7] Beyer, M. A., and Laney, D. (2012). *The importance of 'big data': A definition*. Gartner Research, Stamford, CT, USA, Tech. Rep. G00235055.

[8] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*(Jan), 993-1022.

[9] Blum, A., and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual ACM conference on Computational learning theory*, 92-100.

[10] Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123-140.

[11] Bruce, R. (2001). Semi-supervised learning using prior probabilities and EM. In *International Joint Conference on Artificial Intelligence, Workshop on Text Learning: Beyond Supervision*.

[12] Chapelle, O., Chi, M., and Zien, A. (2006a). A continuation method for semi-supervised SVMs. In *Proceedings of the 23rd ACM International Conference on Machine Learning*, 185-192.

[13] Chapelle, O., Scholkopf, B., and Zien, A. (2006b). *Semi-supervised learning*. MIT Press, Cambridge, MA.

[14] Cozman, F. G., Cohen, I., and Cirelo, M. C. (2003). Semi-supervised learning of mixture models. In *Proceedings of the 20th International Conference on Machine Learning*, 99-106.

[15] Dasarathy, B. V., and Sheela, B. V. (1979). A composite classifier system design: Concepts and methodology. In *Proceedings of the IEEE, 67*(5), 708-713.

[16] Dimitriadou, E., Weingessel, A., and Hornik, K. (2003). *A Cluster Ensembles Framework*. IOS Press, Amsterdam, The Netherlands.

[17] Freund, Y., and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, 148-156.

[18] Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119-139.

[19] Giacinto, G., and Roli, F. (2001). An approach to the automatic design of multiple classifier systems. *Pattern Recognition Letters, 22*(1), 25-33.

[20] Grandvalet, Y., and Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 529-536.

[21] Hansen, L. K., and Salamon, P., (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(10), 993-1001.

[22] Hartley, H. O., and Rao, J. N. (1968). Classification and estimation in analysis of variance problems. *Revue de l'Institut International de Statistique*, 141-147.

[23] Hernández, M. A., and Stolfo, S. J. (1998). Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery, 2*(1), 9-37.

[24] Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning, 42*(1-2), 177-196.

[25] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*(1), 79-87.

[26] Jordan, M. I., and Xu, L. (1995). Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks, 8*(9), 1409-1431.

[27] Kim, S., Zhang, H., Wu, R., and Gong, L. (2011). Dealing with noise in defect prediction. In *IEEE 33rd International Conference on Software Engineering*, 481-490.

[28] Kuncheva, L. I., Bezdek, J. C., and Duin, R. P. (2001). Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition, 34*(2), 299-314.

[29] L'Heureux, A., Grolinger, K., ElYamany, H. F., and Capretz, M. (2017). Machine learning with big data: Challenges and approaches. *IEEE Access*.

[30] Li, M., and Zhou, Z. H. (2005). SETRED: Self-training with editing. In *PAKDD, 3518*, 611-621.

[31] Liu, W., Liu, S., Gu, Q., Chen, X., and Chen, D. (2015). Fecs: A cluster based feature selection method for software fault prediction with noises. In *IEEE 39th Annual Computer Software and Applications Conference (COMPSAC), 2*, 276-281.

[32] Mallapragada, P. K., Jin, R., Jain, A. K., and Liu, Y. (2009). Semiboost: Boosting for semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(11), 2000-2014.

[33] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford coreNLP natural language processing toolkit. In *ACL (System Demonstrations)*, 55-60.

[34] Maulik, U., and Chakraborty, D. (2011). A self-trained ensemble with semisupervised SVM: An application to pixel classification of remote sensing imagery. *Pattern Recognition, 44*(3), 615-623.

[35] Mitra, V., Wang, C. J., and Banerjee, S. (2007). Text classification: A least square support vector machine approach. *Applied Soft Computing, 7*(3), 908-914.

[36] Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2/3), 103-134.

[37] Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine, 6*(30), 21-45.

[38] Provost, F., and Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, Inc..

[39] Riloff, E., Wiebe, J., and Phillips, W. (2005). Exploiting subjectivity classification to improve information extraction. In *Proceedings of the National Conference on Artificial Intelligence, 20*(3), 1106.

[40] Rosenberg, C., Hebert, M., and Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *Seventh IEEE Workshops on Application of Computer Vision, 1*, 29-36.

[41] Sáez, J. A., Galar, M., Luengo, J., and Herrera, F. (2013). Tackling the problem of classification with noisy data using Multiple Classifier Systems: Analysis of the performance and robustness. *Information Sciences, 247*, 1-20.

[42] Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197-227.

[43] Seeger, M. (2000). *Learning with labeled and unlabeled data*. Tech. Rep. Edinburgh, UK: University of Edinburgh.

[44] Tanha, J., van Someren, M., and Afsarmanesh, H. (2011). Disagreement-based co-training. In *23rd IEEE International Conference on Tools with Artificial Intelligence*, 803-810.

[45] Tanha, J., van Someren, M., and Afsarmanesh, H. (2017). Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics, 8*(1), 355-370.

[46] Triguero, I., García, S., and Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems, 42*(2), 245-284.

[47] Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley, Reading, Mass.

[48] Wang, G., Hao, J., Ma, J., and Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications, 38*(1), 223-230.

[49] Wang, L., Chan, K. L., and Zhang, Z. (2003). Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. In *CVPR*, 629-634.

[50] Wang, X. Z., Zhang, S. F., and Zhai, J. H. (2007). A nonlinear integral defined on partition and its application to decision trees. *Soft Computing, 11*(4), 317-321.

[51] Wang, Y., Xu, X., Zhao, H., and Hua, Z. (2010). Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowledge-Based Systems, 23*(6), 547-554.

[52] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[53] Woods, K., Kegelmeyer, W. P., and Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(4), 405-410.

[54] Wu, X. (1996). *Knowledge acquisition from databases*. Ablex Publishing Corp., Norwood, NJ, USA.

[55] Wu, X., and Zhu, X. (2008). Mining with noise knowledge: error-aware data mining. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38*(4), 917-932.

[56] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, 189-196.

[57] Zhou, Z. H. (2012). *Ensemble methods: Foundations and algorithms*. CRC press.

[58] Zhu, X., and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 3*(1), 1-130.

[59] Zhu, X., Lafferty, J., and Rosenfeld, R. (2005). *Semi-supervised learning with graphs*. Doctoral dissertation. School of Computer Science, Language Technologies Institute, Carnegie Mellon University.

[60] Zhu, X., Wu, X., and Chen, Q. (2003). Eliminating class noise in large datasets. In *Proceedings of the 20th International Conference on Machine Learning*, 920-927.

◆ About the Authors ◆

William Xiu Shun Wong

William Xiu Shun Wong is working at the Biz Consulting team of Datasolution Inc. He received the B.S. degree in Computer Science from Universiti Sains Malaysia in 2011 and Ph.D. degree in Business IT from Kookmin University in 2019. His current research interests include text mining, data mining, and text classification.

Donghoon Lee

Donghoon Lee is working at the Data Analysis team of Cafe24 Corp. He is a Ph.D. candidate in the School of Management Information Systems at Kookmin University, Seoul Korea. He received the B.S. degree in Computer Science from Korea National Open University in 2010 and Master degree in Business IT from Kookmin University in 2012. His current research interests include text mining, data mining, and ontology.

Namgyu Kim

Namgyu Kim is a professor in the School of Management Information Systems at Kookmin University, Seoul Korea. He received the B.S. degree in Computer Engineering from Seoul National University in 1998 and Ph.D. degree in Management Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2007. He has been working for Kookmin University since then. His current research interests include text mining, data mining, and data modeling.

Submitted: April 22, 2019; 1st Revision: July 19, 2019; Accepted: September 3, 2019