Master's Thesis

Domain Adaptation for Gender Classification of Text

1610401 WANG Sitong

Supervisor	Kiyoaki Shirai
Main Examiner	Kiyoaki Shirai
Examiners	Hiroyuki Iida
	Minh Le Nguyen
	Shinobu Hasegawa

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

August 2019

Abstract

Due to the rapid growth of Internet, nowadays, people can easily release information or contents to the public. Such user-generated contents often include opinions and emotions of users. Opinion mining is a task to analyze texts written by many users and reveal their opinions toward a specific target such as a product, person or service. It is one of the hot research topics in natural language processing research field. Since different trends of opinions are often found for females and males, it is necessary to distinguish texts written by females or males. Therefore, gender classification, which is a task to identify an author's gender of a given text, is a basic and essential research topic in the opinion mining.

Although there are a large number of studies on the gender classification of Web documents at present, most of them are modeled and predicted in a single type of Web document database. A type of Web documents is often called a "domain" of texts. Blog and microblog (e.g. Twitter, Weibo) are examples of the domain. Most past studies of the gender classification rely on supervised machine learning. However, a model trained in one domain rarely works well in other domains, besides, to prepare labeled datasets of many domains needs enormous costs and time. In this thesis, we focus on the domain adaptation of the gender classification. That is, we aims at building a classification model to identify the author's gender in one source domain and apply it to a different target domain without remarkable loss of the performance. Here, "source" and "target" domains refer to types of documents of the training and test data, respectively.

In the proposed method, a classifier of the gender classification, which judges whether a gender of an author of a text is female or male, is trained from a collection of labeled data. It consists of several steps. The first step is preprocessing. Since texts on the Web are classified in this study, there is much information other than texts, such as URL and non-English words. Such noisy information is removed by simple rules based on regular expression. Then, lemmatization is performed to convert words in inflected forms to base forms. The second step is features extraction. We use word unigram and bi-gram as features for machine learning. The third step is feature selection. For each feature, χ^2 value that evaluates correlation between the feature and the gender class is measured. Then the top 5% or 10,000 features that have the highest χ^2 values are chosen. The last step is training of a gender classification model. A classifier is trained by Naive Bayes or Support Vector Machine (SVM).

In addition to the training of the model of the gender classification, this thesis also considers the domain adaptation, which is the main goal of this study. Among several approaches of the domain adaptation, we focus on two existing domain adaptation methods: the cut-off method and fill-up method. The cut-off method improves the classification accuracy by shortening the feature space of source and target domains by retaining only common features in two domains. On the other hand, the fill-up method extends the feature space of two domains. That is, not only the common features but also domain specific features compose the feature space. Both methods only change the feature space without changing weights of the features. Therefore, domain specific features, which appear only in either the source or target domain, are not heavily considered in a trained classifier. It may lead only a little improvement in the gender classification of different domains.

We propose a novel method for the domain adaptation of the gender classification called "fill-up with word similarity". Although our goal is the gender classification, our proposed method is applicable for any kinds of text classification problems. For a given training data in the source domain and test data in the target domain, we make three sets of features: common features, source specific features and target specific features. For each sample in the source domain, we search target specific features that are similar to one of the features appearing in the sample. Although word uni-gram and bi-gram are used as the features, here we consider only the uni-gram, i.e. word itself. The similarity between words is measured by cosine similarity of two word vectors that are derived from word embedding. We use word embedding obtained by fastText, which is pre-trained from a huge amount of English texts. If the word similarity is greater than the threshold T, similar target specific features are added to the feature vector by changing their weights to 1. Similarly, when a feature vector in the target domain is constructed, similar source specific features are added. In this study, the threshold T is set to 0.7 by our intuition. In our method, source specific and target specific features are taken into account in the training of a gender classifier by changing the weights of similar (or related) domain specific features to 1. It enables us to fill a gap of the feature space between the source and target domains.

Several experiments are conducted to evaluate effectiveness of our proposed method. Two datasets are used: one is Twitter dataset consisting of 260,944 tweets, the other is blog dataset consisting of 268,296 blog articles. These datasets are balanced, i.e. they contain the same number of texts written by females and males. Two cases are considered in this experiment. In the first case, the Twitter dataset is used as the source domain and the blog dataset is used as the target domain. In the second case, two datasets

are swapped. First, it is found that SVM outperforms Naive Bayes, and the feature selection of the top 5% features is better than that of the top 10,000 features. Then, we compare two existing and our proposed domain adaptation methods. The accuracy of the gender classification is 54.64% or 53.00% in the first or second case, when no domain adaptation is applied. The accuracy of the cut-off method is 55.67% or 55.34%, while the accuracy of the fill-up method is 57.73% or 58.55%. Thus the performance of the gender classification is improved by the domain adaptation. Finally, the accuracy of our proposed method is 59.97% or 65.55%. It is better than two baseline methods, especially in the second case. These results prove the effectiveness of our new domain adaptation method.

In future, we should refine preprocessing of Web texts to train an accurate gender classifier. In addition, more sophisticated way to calculate word similarity should be explored.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we firstly explain background of our research in Section 1.1. Section 1.2 describes the motivation and goal of this work as well as the contributions of this thesis. Finally, the structure of the thesis is given in Section 1.3.

1.1 Background

An increasingly large number of people use Internet to express their opinions and release their emotions because of the rapid growth of social networks. In this situation, an unprecedented amount of user-generated data has been produced. It can provide an excellent opportunity for text mining. Opinions and emotions on Web are analyzed for various applications such as opinion mining and reputation analysis. Opinion mining or reputation analysis is a kind of text mining. It aims at revealing users' opinions toward a specific target such as a product, person or service. It provides useful information to not only users but also enterprises that provide products or services. Therefore, opinion mining is paid much attention in recent years.

Authorship analysis is an important aspect of opinion mining, which attempts to know about the author of the web document through many variations about the writing styles that occur between age, gender and social groups. Among various authorship information, this thesis focuses on gender. Since different trends of opinions are often found for males and females, it is required to analyze opinions of males and females separately. However, no information of an author's gender is found for most documents on social media. Therefore, the gender classification of Web texts is essential and critical for precise opinion mining.

Supervised learning is often applied for text classification such as gender

classification. However, it is well known that a classifier obtained by supervised learning heavily dependent on the domain. Here the domain refers to a genre or style of documents. A model trained for one domain (e.g. news domain) seldom works well for other domains (e.g. medical domain). Besides, collecting and curating labeled training sets for different domains is prohibitively expensive, since it requires a lot of human efforts for annotation of correct labels.

"Domain adaptation" or "transfer learning" is a technique to tackle the above problem. It aims to train a robust classifier that can work well for classification on not only the domain of the training data but also other domains. Domain adaptation of the gender classification is essential for opinion mining on Web documents, since there are many domains such as blog, microblog (e.g. Twitter, Weibo), bulletin board system (BBS) and so on. However, the domain adaption of the gender classification is not paid much attention in previous studies.

1.2 Goal

The goal of this thesis is to propose a method of the domain adaptation for the gender classification of an author of a given Web document. We try to build a classification model to identify the author's gender in one source domain and apply it to a different target domain. Our gender classification model can be used to promote applications that can reveal perspective of the opinions of each gender.

The contribution of this thesis is summarized as follows.

- In the past, many researchers have investigated the gender classification or the domain adaptation, but the domain adaptation for the gender classification has not been considered. It is the first attempt to tackle this new research topic.

- We propose a new method named "fill-up with word similarity". It is extension of the existing "fill-up" method. Our method is based on expansion of related features using word embedding.

- We conduct experiments on blog and Twitter datasets to demonstrate that our proposed model outperforms two baseline domain adaptation methods, "cut-off" and "fill-up".

1.3 Thesis Outline

The rest of this thesis is organized as follows.

- Chapter 2 describes related work about gender classification, domain adaptation, Support Vector Machine(SVM), and word embedding.
- Chapter 3 first illustrates the task of gender classification of text. Then we explain details of our proposed method of the domain adaptation for the gender classification. In addition, we also describe two existing methods of the domain adaptation for comparison with the proposed method.
- Chapter 4 reports results of experiments on corpora in two different domains: blog domain and Twitter domain. Furthermore, we also conduct an error analysis and show major causes of errors such as difficulty of preprocessing of Web documents.
- Chapter 5 concludes this study and denotes some future work to improve the proposed method.

Chapter 2

Related Work

This chapter consists of 4 sections. Section 2.1 introduces related work of the gender classification. Section 2.2 introduces some related work of the domain adaptation. Section 2.3 presents a powerful and boost supervised learning algorithm called Support Vector Machine (SVM), since it is used for the gender classification in this study. Section 2.4 explains word embedding that plays an important role in the proposed method. Finally, Section 2.5 clarifies the difference between the previous studies and the proposed method.

2.1 Gender Classification

Gender classification is a task to identify gender of an author of a given document, where the gender is usually defined as a female or male. As discussed in Section 1.1, the gender classification is important for the opining mining. Several studies have already made for the gender classification.

Yan et al. present a Naive Bayes classification method to identify genders of webloger that post their opinion to a blog platform[20]. In addition to features employed in traditional text categorization such as word uni-gram, they also use weblog-specific features such as website profile background colors and emoticons. As for the feature selection, they make a short list of the uni-gram features with the high mutual information with the gender class. They carry out an experiment using 75,000 personal blog texts posted by 3,000 bloggers excerpted from a free blog service called Xanga. In Xanga, a huge number of users actively post blog articles, while their genders are shown in their profile. According to their experiments, Naive Bayes achieves the best performance. Its precision, recall and F-measure are 65%, 71% and 68%, which are 15, 17, and 18 points higher than the baseline, respectively.

Mukherjee and Liu propose two new techniques to improve the accuracy

of the gender classification of microblog writers[12]. The first one is proposal of new features for this task: variable length POS patterns, stylistic features that reflect people's writing style, and gender preferential features (e.g. women like to use insensitive words such as "lucky" and "so" more than men). The second one is to propose an ensemble feature selection method which use many different types of feature selection criteria. Their techniques achieve 88.56% accuracy, which is around 9 points better than other methods.

Burger et al. investigate high performance classifiers for identifying the gender of Twitter users[2]. In their method, not only user's tweets but also other metadata such as a screen name, full name, and description are used to determine a gender of a Twitter user. Word and character n-gram are extracted from these four kinds of texts as features for machine learning. To construct a set of Twitter users labeled with their gender, a URL to a user's blog site in Twitter profile is followed, then a gender field in a blogger profile is used to identify the gender of the user. In the preliminary experiment, Support Vector Machine (SVM), Naive Bayes, and Balanced Winnow2 are compared using only word uni-gram features, and find that Balanced Winnow2, whose accuracy is 74%, is the best. Then, a gender classifier is trained by Balanced Winnow2 using their proposed features. Its accuracy is 92%.

Mansur and Wolfe focus on the gender classification of authors of small texts of mobile/web application reviews, and propose a method to train SVM with the word-based stylometric features[9]. Here the word-based stylometric features are 399 function words, which specify the attitude or mood of the writer. They train SVM from the Enron email data including 93,265 messages and get 93% accuracy when the authors in the test data also appear in the training data.

2.2 Domain adaptation

Although the domain adaptation has been introduced in Section 1.1, we here explain more detailed background and problems of the domain adaptation. Although supervised machine learning is successfully applied for many tasks on natural language processing, it requires labeled data for training a classifier. Construction of labeled data requires a lot of efforts due to manual annotation of texts with gold labels. On the other hand, there exists a wide variety of texts in terms of topics and/or writing styles. It is impossible to make labeled data for all kinds of texts. If there is no labeled data for a certain test data, a classifier is trained from existing labeled data on different texts. We often assume that texts in a training and test set are consistent in

supervised learning, but in such cases it is not true.

When the textual characteristics of a training and test set are inconsistent, the model trained according to the minimum empirical error criterion in the training data has poor performance on the test data due to over-fitting. The domain adaptation or transfer learning is a technique to train a robust classifier so that it considerably works well even when it is applied to a different type of texts.

There are two key concepts in the domain adaptation: source domain and target domain. Source domain represents a type of texts of a training data with supervisory information. Target domain represents a domain where test samples are located. It is assumed that data in a target domain is not labeled or labeled for only a limited number of samples. In the domain adaptation, source and target domains are assumed to have different textural characteristics.

1. **Sample adaptation**[19]
   In this approach, weights of samples in the source domain are adjusted so that a distribution of features in the source domain becomes close to the target domain.

2. **Feature level adaptation**[8]
   In this approach, the source domain and the target domain are projected to a common feature space.

3. **Parameter transfer adaptation**[13]
   Different knowledge can be transferred through tasks since the transferred knowledge is encoded into the shared parameters or priors.

In this situation, many researchers have tried to come up with a huge number of methods for this task in the past as follows.

Duame and Marcu introduce a powerful domain adaptation method based on probability distribution[6]. It is a production model that contains domain-specific distribution and common distribution among domains. They use CEM algorithm to estimate parameters. Compared with traditional methods, they can achieve good results. However, the complexity of the algorithm is high.

Chen et al. propose an algorithm named CODA (Co-Training for Domain Adaptation) that bridges the gap between source and target domains by slowly adding target features and instances in which the current algorithm is the most confident[4]. They apply CODA for training a classification model that judges if a review for a product is positive (higher than 3 stars) or negative (3 stars or lower). They evaluated their method on the "Amazon

reviews" benchmark data sets including reviews of four different types of products: books, DVDs, electronics, and kitchen appliances.

2.3 Support Vector Machine

Support Vector Machine (SVM)[7] is a pattern recognition method based on statistical learning theory, which is widely used in solving classification problems. SVM model determines the best hyperplanes or lines for separating the several classes in high dimensional feature space. It is known that SVM relatively performs well when the size of training data is small and the number of features is extremely high.

SVM is widely used for text classification. The goal of the text classification is to classify a given text into one of several predefined categories. SVM is based on the Structural Risk Minimization principle[18] and it really has substantially improved the previous methods and performed well in various text classification tasks. In addition, because an SVM classifier can be well generalized in high-dimensional feature space, there is no need for feature selection, which makes the application of text classification easier. However, we try feature selection in the gender classification even when we use SVM, since we believe feature selection is still effective. Furthermore, training of SVM is fully automatic and manual parameter adjustment is not required, which means it can be used easily and efficiently.

In this study, we use LIBSVM[3] which is developed by Professor Chih-Jen Lin of Taiwan University as a tool for training SVM. LIBSVM can be used for both classification (supporting binary classification and multi-class classification) and regression. It has preferable characteristics of simple operation, easy to use, fast and effective, and relatively little adjustment to the parameters involved of SVM. The types of SVM supported in LIBSVM are: C-SVC (multi-class classification), nu-SVC (multi-class classification), one-class SVM, epsilon-SVR (regression) and nu-SVR (regression). The types of kernels supported by LIBSVM are linear, polynomial, radial basis, sigmoid and precomputed kernel (kernel values in training set file). Although there are many parameters needed in LIBSVM, most of them have their default values.

The data set is usually scaled when LIBSVM is used. The purpose of scaling is to prevent some eigenvalue ranges from being too large and others too small, and to reduce computational costs to calculate inner product of two vectors for calculating kernel function in training. Weights in feature vectors is usually scaled between $[-1, 1]$ or $[0, 1]$. In this study, we always set weights of features as 0 or 1.

2.4 Word embedding

In the method of the domain adaptation proposed in this thesis, word embedding is used and plays an important role. This section introduces what word embedding is and how it is used in general.

Word embedding is a term in natural language processing field, which means a vector representation of words or phrases in vocabulary. Specifically, word embedding is dense vectors that represent words. The locations of words in vector space are acquired from the contextual similarity of words in a huge amount of texts. There are several methods to train word embedding such as Skip-gram model[11], Glove[15] and fastText[1]. Word embedding can be trained by (relatively small) target corpus or pre-trained from a large amount of general texts. Word embedding is a kind of powerful and great distributed vector representation which can capture not only precise syntactic but also semantic properties of words. It helps machine learning models, especially deep learning models, to achieve good performance in natural language processing tasks by its ability to capture word similarity.

The vigorous development of word embedding has led to a large number of in-depth studies. Researchers explored how to improve the performance of word embedding in different tasks. Word embedding can be used for various natural language processing tasks. Several examples of the usage of word embedding are shown below[21].

- Similarity between words
  In natural language processing, it is often required to measure the similarity between two words. Word embedding can be used to measure the word similarity. Since word embedding is vector representation of words, any indices to measure similarity between two vectors can be defined as word similarity. The most frequently used index is cosine similarity, which is defined as Equation (2.1). Cosine similarity represents cosine of an angle between two word vectors as indicated in Figure 2.1. It is in a range of [−1,1]. The greater the value is, the higher the similarity between the i_{th} and the j_{th} words is. In the Figure 2.1, the angle between "girl" and "boy" is small, while the angle between "girl" and "eye" is large. Note that cos θ becomes great when the angle θ becomes small.

$$\cos(v_i, v_j) = \frac{v_i \cdot v_j}{||v_i|| * ||v_j||} \tag{2.1}$$

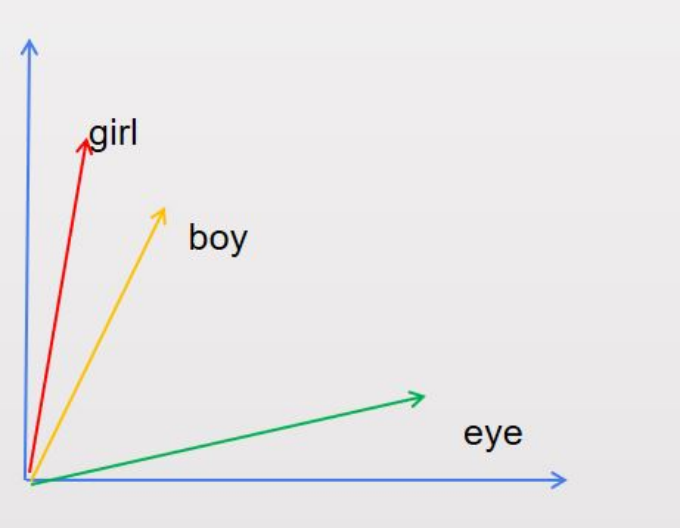

Figure 2.1: Geometric Similarity of Word Vectors

- Analogy
  Here the analogy examines whether the two pairs of words have the same relation. Mikolov et al. showed some examples of relations such as king-queen = man-woman (this relation is words of different genders), walk-walking = run-running (this relation is a base and progressive form of a verb), Paris-France = Berlin-Germany (this relation is a capital city and country), and so on[11]. Supposing the relation A-B = C-D, the word D is guessed when the three words A, B, C are given.

  The geometric property corresponding to the semantic property of analogy is parallelism: whether the difference of two pairs of word vectors is parallel to each other, as shown in the Figure 2.2. More specifically, a vector of the word D is synthesized as $\vec{D} = \vec{B} - \vec{A} + \vec{C}$, then a word whose word embedding is the most similar to $\vec{D}$ is obtained as the word D.

- Clustering
  Clustering of words is a task to make clusters (groups) of words. In the most of clustering algorithms, each data sample is represented as a vector, and samples are merged into a single cluster when the vectors of words are geometrically close. Since word embedding provides vector representation of words, it can be applied for clustering straightforwardly. Figure 2.3 shows an example of clustering of words. In this figure, “dog”, “cat” and “mouse” are merged. All these words represent animals.

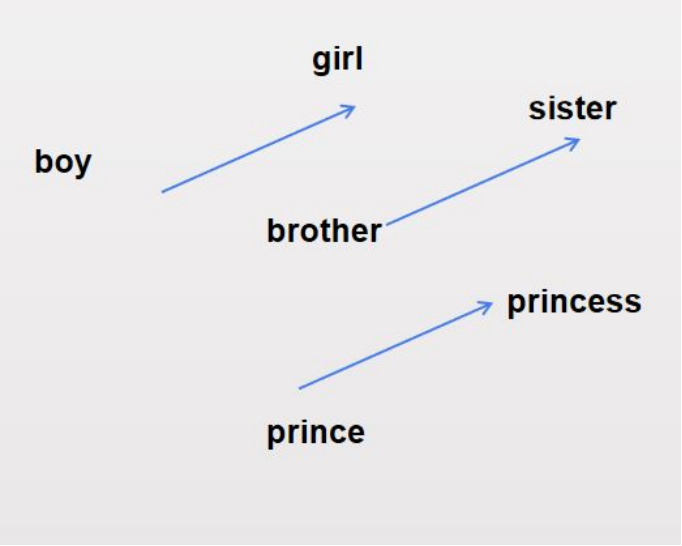

Figure 2.2: Analogy of Word Vectors

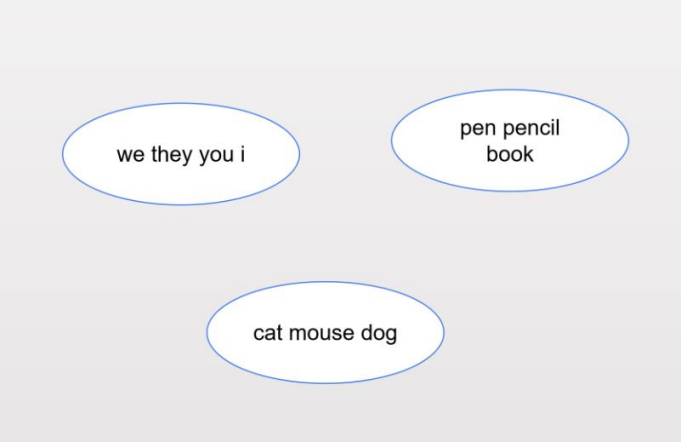

Figure 2.3: Clustering of Word Vectors

2.5 Characteristic of this study

In the past, many researchers have investigated gender classification as discussed in Section 2.1 or domain adaptation as discussed in Section 2.2, but domain adaptation for gender classification has not been considered. This thesis empirically investigate how difference of domains of the training and test data influences the accuracy of the gender classification, and how effectively the existing domain adaptation can alleviate this problem. In addition, we propose a new domain adaptation method named "fill-up with word similarity", which can achieve better accuracy than two baseline methods: the "cut-off" method and "fill-up" method.

Chapter 3

Proposed Method

3.1 Task Definition

The task considered in this study consists of two subtasks. One is gender classification and the other is domain adaptation. In this section, we will explain the details of these subtasks.

Gender classification is a task to determine a personal gender, such as male or female, of an author of a given text. In general, the gender of an author is guessed based on his or her characteristic. The result of the gender classification is often a binary value such as 1 or 0, representing either male or female. That is, the gender classification is essentially a binary classification problem. In this study, the gender classification is experimented on Web documents.

Domain adaptation, which is also known as "transfer learning", is a popular but challenging task. It is a research topic associated with machine learning[13]. In general, machine learning poorly performs when the domains of the training and test data are different. Domain adaptation aims to build classification models that are robust to mismatched characteristics of the texts in the training and test domains. In the domain adaptation, the domain of the training data is called "source domain", while the domain of the test data is called "target domain".

According to the homogeneity of feature space in different source and target domains, the domain adaptation is divided into two types: homogeneity adaptation and heterogeneity adaptation. In the homogeneous domain adaptation, the feature space of the two domains is the same, but the feature distribution is different. In the heterogeneity domain adaptation, on the other hand, the feature space of two domains are different. Compared with the isomorphic domains, the heterogeneity of the feature space makes

the domain adaptation more difficult. In fact, there are many cases where the feature space of two domains are not completely heterogeneous. Specifically, two domains share some common characteristics (or features), while each domain has its own domain specific characteristics. This can be regarded as a special case of heterogeneous domain adaptation, where there exists both common characteristics and domain specific characteristics. We call this problem as a hybrid domain adaptation.

This study focuses on a task of the hybrid domain adaptation for the gender classification. Blog and Twitter are considered as the source and target domain, and vice versa. With several supervised learning algorithms, we try to build a classification model to identify the author's gender in one source domain and apply it to a different target domain.

3.2 Gender Classification

This section describes the gender classification task. That is, we explain the way how to train a classifier for the gender classification. It consists of several steps: data preprocessing, feature extraction, feature selection and training of a classifier. Figure 3.1 illustrates how a classifier of the gender classification is obtained from a collection of texts labeled with gold gender. The following subsections describe the details of these steps.

3.2.1 Preprocessing

This thesis aims at the gender classification of English texts. Therefore, it is necessary to first remove all of the non-English words including punctuation (eg. ',', '"', '!', '.'), number and emoticon. These non-English words are eliminated by pattern matching with regular expression[10]. In addition, if a sentence includes a website address, it is replaced with a special token "urllink". For instance, the sentence S is converted to a simplified sentence S1 as follows.

S= These are some smiling students :-) :-P. click https://blog.csdn.net/cai
S1= These are some smiling students click urllink

After removing non-English words, a sentence is tokenized. Tokenization is a process to divide a sentence into a sequence of words. There are many ways of tokenization currently, but in this study, the Natural Lan-

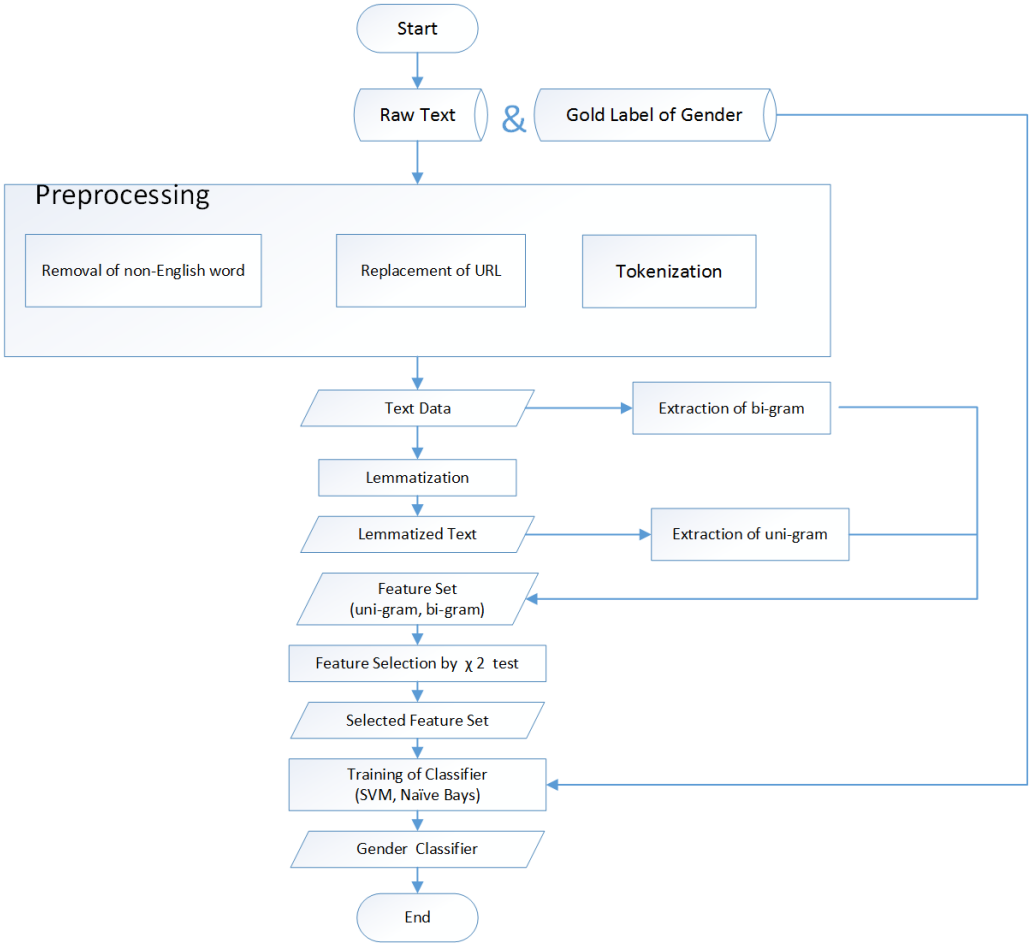

Figure 3.1: Flowchart of gender classification

guage Toolkit (NLTK)[1] tokenizer is used. It divides a string into substrings by splitting on the specified characters such as a period and comma. For instance, the sentence S1 is tokenized as the sentence S2 as follows.

S1= These are some smiling students click urllink
S2=['these', 'are', 'some', 'smiling', 'students', 'click', 'urllink']

3.2.2 Feature extraction

The next step is feature extraction. In this study, word n-gram is used as features for training of the gender classifier. Since word n-gram is widely used in natural language processing, we briefly introduce it. Word n-gram is a sequence of consecutive n words in a sentence. When n is 1, 2 or 3, word n-gram is called uni-gram, bi-gram or tri-gram. Word uni-gram is also known as bag-of-words. A set of word n-gram is commonly used as simple representation of a sentence. For example, it can be used to measure the similarity between two sentences. Even when two sentences are not exactly the same, they are regarded as similar if they contain many common word n-gram. This is something like fuzzy matching of sentences. Word n-gram is simple, powerful and robust. It is often observed that a model using simple features (such as word n-gram) trained from huge data is better than a model using complex features trained from small data. Therefore, word n-gram is a widely used feature for machine learning.

In this study, from a given sentence, word uni-gram and bi-gram are extracted as the features. To extract word uni-gram feature, lemmatization is performed. In many languages, words appear in several inflected forms. For example, in English, the verb 'to work' may appear as 'work', 'worked', 'works', and 'working'. The base form (e.g. 'work') , which is often used as a headword in a dictionary, is called a lemma of a word. Lemmatization in linguistics is the process of grouping together the inflected forms of a word so that they can be analysed as a single item. That is, a lemma is used as a canonical form of an inflectional word. The NLTK is used for lemmetization in this study. The sentence S2 is converted by lemmetization as the sentence S3 as follows. Note that 'are', 'smiling', and 'students' are converted into their base forms as 'be', 'simile', and 'student', respectively. S3 is also a set of extracted uni-gram.

[1]https://www.nltk.org/

S2=[‘these’, ‘are’, ‘some’, ‘smiling’, ‘students’, ‘click’, ‘urllink’]
S3=[‘these’, ‘be’, ‘some’, ‘smile’, ‘student’, ‘click’, ‘urllink’]

Word bi-gram is also extracted. That is, all two adjacent words in a sentence are extracted as the features. Unlike uni-gram, lemmatization is not applied for extraction of word bi-gram. In other words, forms of all words are kept as in a sentence whether it is a base form or inflected form. This is because inflection makes sense in word bi-gram. For example, “barking dog” and “barked dog” are word bi-gram that have different meanings. For example, the sentence S2 is converted to a list of bi-gram as the sentence S3’ as follows.

S2=[‘these’, ‘are’, ‘some’, ‘smiling’, ‘students’, ‘click’, ‘urllink’]
S3’=[‘these-are’, ‘are-some’, ‘some-smiling’, ‘smiling-students’, ‘students-click’, ‘click-urllink’]

As a consequence of extraction of word uni-gram and bi-gram, the sentence S2 is converted to the sentence S4 as follows.

S2=[‘these’, ‘are’, ‘some’, ‘smiling’, ‘students’, ‘click’, ‘urllink’]
S4 = [‘these’, ‘be’, ‘some’, ‘smile’, ‘student’, ‘click’, urllink’, ‘these-are’, ‘are-some’, ‘some-smiling’, ‘smiling-students’, ‘students-click’, ‘click-urllink’]

3.2.3 Feature Selection

Feature selection is a process to choose effective features among a set of feature candidates to improve the classification performance. In this study, it means to select the word uni-gram and bi-gram features that are effective for the gender classification.

χ^2 test method [5] is used for the feature selection. It is based on a statistical hypothesis test where the sampling distribution of the test statistic is a χ^2 distribution when the null hypothesis is true and it is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies of the features in data of two classes, namely texts written by males and females. χ^2 value measures how much expected counts

and observed counts deviate from each other as defined in Equation (3.1).

$$\chi^2(f,c) = \frac{N(WZ - YX)^2}{(W+Y)(X+Z)(W+Z)(Y+Z)} \tag{3.1}$$

χ^2 value measures the correlation between the feature f and the gender class c. The first step in computing the χ^2 value is the computation of the contingency table. Table 3.1 shows a 2×2 contingency table for χ^2 test. f or $\bar{f}$ represents that the feature is presented or not presented in the text, while c or $\bar{c}$ represents that the class of the text is c or not c. In this study, c denotes the class of female and $\bar{c}$ denotes the class of male. W, X, Y, and Z denote the frequency of each case. W denotes the frequency of texts that includes the feature and belong to the females class. X denotes the frequency of texts that include the feature and belong to the male class. Y denotes the frequency of texts that do not include the feature and belong to the female class. Y denotes the frequency of texts that do not include the feature and belong to the male class. Finally, N is the sum of W, X, Y and Z, which is equivalent to the total number of texts in the data.

Table 3.1: 2×2 contingency table of feature(f) and class(c)

	c	$\bar{c}$
f	W	X
$\bar{f}$	Y	Z

We can use χ^2 values to determine whether features are highly related to the gender classes. The larger the χ^2 value is, the stronger the correlation between the feature and the gender class is. The smaller the χ^2 value is, the more irrelevant the feature is. Features with high χ^2 value should be preserved, while ones with low χ^2 value should be removed. We set a threshold to control the number of selected features. We use two kinds of thresholds, T_1 and T_2. T_1 is the proportion of the number of selected features to the total number of features. T_2 is the number of selected features. By our intuition, we set T_1 to 5% and T_2 to 10,000. It means that only the top 5% or 10,000 features with the highest χ^2 values are chosen.

Let us show an example of the feature selection. Now we suppose that we try to select the top 5% features (T_1=5%) and χ^2 values of the top 5% features are more than 2.1862. Table 3.2 shows features and their χ^2 values. By removing several features with low χ^2 values, the feature list of the sentence S4 is converted to S5.

S4 = [‘these’, ‘be’, ‘some’, ‘smile’, ‘student’, ‘click’, urllink’, ‘these-are’, ‘are-some’, ‘some-smiling’, ‘smiling-students’, ‘students-click’, ‘click-urllink’]

S5 =[‘these’, ‘be’, ‘some’, ‘smile’, ‘student’, ‘click’, urllink’, ‘these-are’]

Table 3.2: Example of features and χ^2 values

Feature	χ^2 value
these	33.3152
be	321.7248
some	42.5940
smile	52.9841
student	3.9699
click	172.2771
urllink	1605.0916

Feature	χ^2 value
these-are	4.9276
are-some	0.0905
some-smiling	0.7449
smiling-students	1.0931
students-click	0.7232
click-urllink	1.0247

3.2.4 Training of Classifier

After the feature extraction and feature selection, a classifier that determines the gender class of a given text is obtained by supervised machine learning. In this study, LIBSVM[2] is used to train SVM classifer. It is an integrated software for support vector classification, regression, and distribution estimation. It supports 5 types of SVM: C-SVC, nu-SVC, one-class SVM, epsilon-SVR and nu-SVR. C-SVM is chosen in this study. It is a default SVM in LIBSVM tool. There are also 4 types of kernel functions: linear function, polynomial function, radial basis function and sigmoid function.

There are several requirements to use LIBSVM. Firstly, each data should be represented as a set of its class and features in one line. In addition, a weight of each feature should also be denoted. The data format in LIBSVM is as follows:

Class1 Feature11:Weight11 Feature12:Weight12 ...
Class2 Feature21:Weight21 Feature22:Weight22 ...

[2]https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Secondly, all of features should be represented by integers in LIBSVM. Therefore, it is necessary to convert a class and feature into an unique number. For example, the sentence S5 represented as the feature list is converted to the sentence S6' as follows.

S5=['these', 'be', 'some', 'smile', 'student', 'click', urllink', 'these-are']
S6'=['11', '34', '35', '36', '37', '38', '39', '200']

Thirdly, all classes should be represented by integers in LIBSVM. Therefore, the 'female' and 'male' classes are converted to '1' and '2', respectively.

Fourthly, the weight of the feature should be represented by a real number. Although there are several methods to define the weights of features for machine learning, the most simple method is used in this study. That is, the weights of all features presented in a sentence is defined as 1.

Let us show an example of data conversion. The sentence S5 that belongs to the female class is converted into the S6 that is acceptable in LIBSVM tool.

S5=['these', 'be', 'some', 'smile', 'student', 'click', 'urllink', 'these-are']
S6= 1 11:1 34:1 35:1 36:1 37:1 38:1 39:1 200:1

3.3 Domain Adaptation

The main goal of this thesis is to propose a novel method of the domain adaptation for the gender classification. This section describes three methods of the domain adaptation: "cut-off"[14], "fill-up"[14], and "fill-up with word similarity". Only "fill-up with word similarity" is our proposal; "cut-off" and "fill-up" are the baseline methods. The details of these two baselines are also explained in this section to clarity the difference between our method and the baselines.

In the domain adaptation, it is assumed that there are two datasets of a source domain (used as the training data) and a target domain (used as the test data). When the source domain and the target domain are different, the features in two data sets are also different. Hereafter, we define three kinds of features as follows.

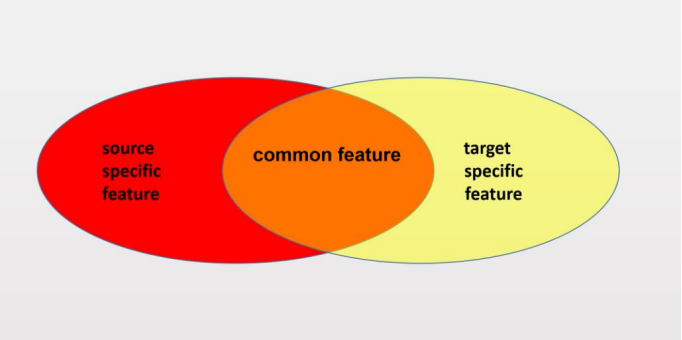

Figure 3.2: Feature space in the source and target domains

common feature
: A feature that appears in the datasets of both the source and target domains.

source specific feature
: A feature that appears in the source domain, but not in the target domain.

target specific feature
: A feature that appears in the target domain, but not in the source domain.

The graphical representation of these features are shown in Figure 3.2. Note that a gap between the feature spaces between the source and target domains is a major cause of decline of the classification performance. There might be many target specific features that are effective for the gender classification in the target domain, but such features do not appear in the source domain (i.e. in the training data). Thus these features are ignored in the trained classifier. Our main idea of the domain adaptation is to close up a gap of two feature spaces.

3.3.1 Cut-off method

The cut-off method is a straightforward way for aligning the feature spaces by cutting off the specific features in each domain and preserving only common features. For each data in the source domain, source specific features are removed and common features are preserved. Similarly, for each data in the target domain, target specific features are removed and common features are preserved. In this way, the feature spaces between the source and target

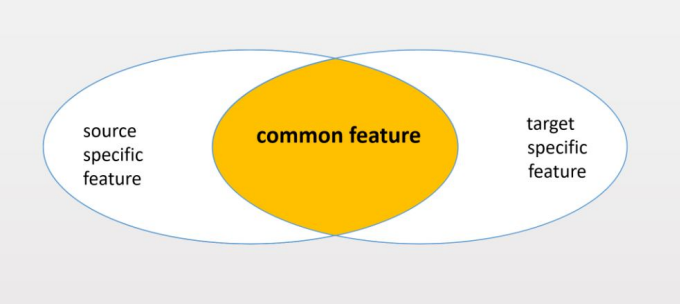

Figure 3.3: Feature space of the cut-off method

domain are unified. The feature space in the cut-off method is shown in Figure 3.3.

Let us show an example of modification of feature vectors by the cut-off method. S7 is a sentence in the source domain, while S8 is a sentence in the target domain. When only "like" and "and" are common features, only these features are kept. S7' and S8' are modified feature vector of S7 and S8.

S7 = male posa like poeticism, poplin, and poundcake
S8 = female i like apple and dog
S7' = male like:1 and:1
S8' = female like:1 and:1

3.3.2 Fill-up method

The fill-up method is another method for the domain adaptation. Unlike the cut-off method that shortens the feature space, the fill-up method expand the feature space including common, source specific and target specific features. It can preserve the discriminative information for each domain which is essential for the domain adaptation. For each data in the source domain, all target specific features are added while the weights of the added target specific features are set to zero. It means that the number of features whose weights are greater than zero is not changed, but only the size of the feature vector is enlarged. Similarly, for each data in the target domain, all source specific features are added with zero weights. In this way, the feature spaces of the source and target domains are unified as a set of all features including common, source specific and target specific features. Figure 3.4 represents the feature space in the fill-up method.

Let us show an example of modification of feature vectors by the fill-up method. S7 and S8 are example sentences in the source and target domains respectively, which are the same as ones shown in Subsection 3.3.1 (Cut-off

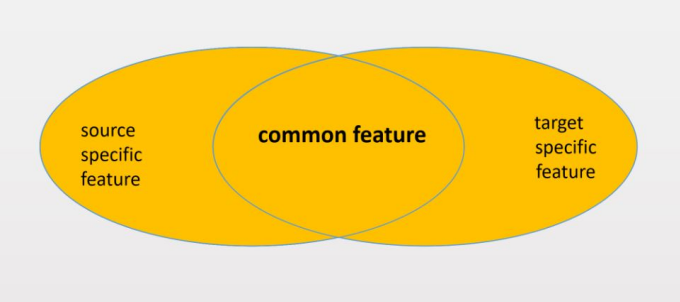

Figure 3.4: Feature space in the fill-up method

method). Note that it is assumed that only "like" and "and" are the common features. S7 is converted to $S7^l+$ by adding the target specific features and setting their weights to zero. Note that actually all target specific features appearing in the data of the target domain are added. Similarly, S8 is converted to $S8^+$ by adding the source specific features with zero weights.

S7 = male posa like poeticism, poplin, and poundcake
S8 = female i like apple and dog
$S7^+$ = male posa:1 like:1 poeticism:1 poplin:1 and:1 poundcake:1 i:0 apple:0 dog:0
$S8^+$ = female i:1 like:1 apple:1 and:1 dog:1 poeticism:0 poplin:0 poundcake:0

3.3.3 Fill-up with word similarity

Our proposed method is called "fill-up with word similarity". Similar to the fill-up method, it also expands the feature space for each domain by concatenating common, source specific and target specific features. In addition, when a feature vector of a sentence in a source domain is expanded, weights of relevant target specific features are set to 1. More specifically, if the word similarity between a target specific feature and one of features that appear in a sentence is greater than a certain threshold T, the weight of the target specific feature is set to 1. Similarly, when a feature vector of a sentence in a target domain is expanded, weights of 'similar' source specific features are set to 1. The feature space made by the fill-up with word similarity method is shown in Figure 3.5. The importance difference between the fill-up method and our method is as follows. In the fill-up method, the feature spaces in the source and target domain are unified, but the source and target specific features are not heavily considered because those weights are always set to zero. In the fill-up with word similarity method, the relevant source specific features or target specific features are explicitly added to the feature vector

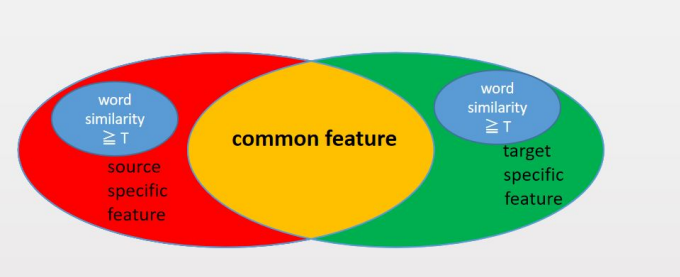

Figure 3.5: Feature space of the fill-up with word similarity method

by setting their weights to 1. It enable us to close up a gap of the feature spaces between the source and target domain more densely.

In our method, how to calculate the similarity between two words is a critical point. Word embedding is used for calculation of word similarity. Word embedding is a technique for converting words into vectors or matrix forms that computers can easily handle. It is also a kind of powerful and great distributed vector representation of words, which can capture not only precise syntactic but also semantic properties of words[17]. We use word embedding obtained by a tool called fastText[3]. More specifically, we use the pre-trained word embedding obtained from Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset. The word similarity between two words is defined as cosine similarity of two word vectors given by word embedding as in Equation (2.1).

The threshold of word similarity T is also the important parameter. It controls the number of domain specific features to be added to the feature vector. When T is too small, many irrelevant features may be added. It may cause decline of accuracy of the gender classification. On the other hand, when T is too large, only a few domain specific features are added, and a gap of the feature spaces of two domains may not be filled sufficiently. In this study, T is determined as 0.7 by the preliminary experiment. Note that we use word uni-gram and bi-gram as the features, but only word uni-gram is considered for the feature expansion. Because the similarity between uni-gram features can be measured by word embedding, but the similarity between bi-gram features cannot. Domain specific bi-gram features are also added to a feature vector, but their weights are always set to zero as in the fill-up method.

Algorithm 1 is the pseudocode to determine a set of new features F_{sn} for a given sentence in a source domain. In other words, F_{sn} is a set of the target specific features whose weights are set to 1. Let us suppose that F_c is a set

[3]https://github.com/facebookresearch/fastText

of the common features, F_s is a set of the source specific features, F_t is a set of the target specific features[4]. T is the threshold of the word similarity. For each source specific feature f_i, we check if the weight of f_i is 1, i.e. f_i appears in the sentence. Note that $w(f_i)$ at line 3 stands for a weight of a feature f_i. Then, for each target specific feature f_j, the word similarity between f_i and f_j, sim(f_i, f_j) at line 5, is calculated. If it is greater than T, f_j is added to F_{sn}. The weights of the features in F_{sn} are set to 1 in the modified feature vector.

Algorithm1: fill-up with word similarity in source domain

Data: $F_c = \{f_{c1}, f_{c2}, \ldots f_{cl}\}$,
$F_s = \{f_{s1}, f_{s2}, \ldots f_{sm}\}$,
$F_t = \{f_{t1}, f_{t2}, \ldots f_{tn}\}$,
T.

Result: F_{sn}

```
1   F_sn ∈ ∅
2       for each f_i in F_s
3         if w(f_i) = 1    then
4            for each f_j in F_t
5               if sim(f_i, f_j) ≥ T    then
6                  F_sn ← f_j
7               end if
8            end for
9         end if
10      end for
```

[4]Word bi-gram features are omitted in Algorithm 1. Therefore, F_c, F_s, and F_t contain only word uni-gram features.

Algorithm 2 is the pseudocode to determine a set of new features F_{tn} for a given sentence in a target domain. It is almost the same as Algorithm 1. In this case, source specific features are added to F_{tn} when the word similarity between it and the feature in the sentence is high. The weights of the features in F_{tn} are set to 1 in the modified feature vector.

Algorithm2: fill-up with word similarity in target domain
Data: $F_c = \{f_{c1}, f_{c2}, \ldots f_{ci}\}$,
$F_s = \{f_{s1}, f_{s2}, \ldots f_{si}\}$,
$F_t = \{f_{t1}, f_{t2}, \ldots f_{ti}\}$,
T.
Result:F_{tn}

```
1   F_tn ∈ ∅
2      for each f_i in F_t
3        if w(f_i) = 1 then
4           for each f_j in F_s
5              if sim(f_i, f_j) ≥ T then
6                 F_tn ← f_j
7              end if
8           end for
9        end if
10     end for
```

The example of similar words derived from word embedding obtained by fastText is shown in Table 3.3. "poundcake" is a word in a source domain, while "bhajis", "damson" and "brique" are words that are the most similar to "poundcake"in a target domain. Among these three words, only "bhajis" is added to F_{sn} since the similarity is greater than T(=0.7). Here, the sentence S9 in a source domain is converted into S10 by our proposed method as follows. Note that the similar words of "posa", "poeticism" and "poplin" are also added and their weights are set to 1 in S10.

S9 = male posa like poeticism, poplin, and poundcake

S10 = male posa:1 verga:1 sincero:1 territori:1 listo:1 menta:1 acabo:1 vinte:1 venta:1 hieu:1 konta:1 like:1 poeticism:1 weltschmerz:1 poplin:1 batiste:1 poundcake:1 bhajis:1

Table 3.3: Example of similar words

word	most similar words in target domain
posa	verga:0.7586859 sincero:0.7336019 territori:0.7218477 listo:0.71290433 menta:0.7105021 acabo:0.710363 vinte:0.7085628 venta:0.7047338 hieu:0.7021778 konta:0.70144004 tanta:0.6975446 museu:0.69355106
poeticism	weltschmerz:0.7078168 unrepressed:0.6533393 crowd-pleaser:0.620426
poplin	batiste:0.7304473 jacquard:0.6781405 chemises:0.6774645
poundcake	bhajis:0.7035682 damson:0.662533 brique:0.5953913

Chapter 4

Evaluation

4.1 Data

We evaluate our proposed models of the domain adaptation for the gender classification of texts on two benchmark datasets that are collections of the texts annotated with the authors' gender. One is a dataset of Twitter, the other is one of blog texts. In other words, Twitter and blogs are regarded as two different domains in this experiment. We introduce them one by one in the following subsections.

4.1.1 Twitter dataset

The Twitter dataset is made by an online community of data scientists and machine learners named kaggle[1]. It includes a quite big collection of text labeled with the self-provided gender. It is 350 MB data consisting of about 3.8 million tweets. In this experiment, we just choose 260,944 tweets with an equal number of texts written by females and males. That is, the Twitter dataset is balanced.

4.1.2 Blog dataset

The blog dataset is named "The Blog Authorship Corpus"[16] that consists of the collected posts of 19,320 bloggers gathered from blogger.com. It includes 681,288 posts and over 140 million words, i.e. almost 35 posts and 7,250 words per person. Each blog article is labeled with a blogger id as well as the blogger's age, industry, gender, and astrological sign. In this study, to make the amount of Twitter and blog datasets be almost the same, we use

[1]https://www.kaggle.com/s1m0n38/twitter-text-and-gender/version/1

Table 4.1: Statistics of dataset

Datasets	Twitter	Blog
female	130,472	134,248
male	130,472	134,248
total	260,944	268,296
training data	234,850	241,474
test data	26,094	26,822

the balanced 268,296 blog articles with the same number of texts written by females and males.

4.1.3 Statistics of the datasets

Table 4.1 shows statistics of Twitter and blog data. As already explained, the number of samples of two datasets are equivalent, and they consists of the same numbers of female and male texts. Firstly, for every classification task, the dataset must be divided into two types of data: training data and test data. In this study, the dataset of either domain is divided into a training and test data. As shown in Table 4.1, 90% and 10% texts are used as the training and test data, respectively. Then, these data are used in two different ways. One is the gender classification where the source and target domains are the same. In this experiment, the training data of Twitter (or blog) is used for training of a classification model, and the test data of Twitter (or blog) is used for evaluation of the trained model. The other is the gender classification where the source and target domains are different, which focuses on evaluation of domain adaptation methods. In this case, a classifier is trained from the training data of Twitter (or blog) and is applied for the classification of the test data of blog (or Twitter).

4.2 Evaluation criterion

In the experiment, the performance of the gender classification with and without the domain adaptation should be measured. The gender classification is a kind of text classification. In general, the performance of text classification is mainly manifested in two aspects: classification efficiency and classification effectiveness. The classification efficiency refers to the training and classification time of classifiers; the classification effectiveness refers to the ability of classifiers to make correct decisions. Specifically, the evaluation index of the classification efficiency is time, that is, the training time of the classifier

and the time required for classification of a single document. The evaluation index of classification effectiveness is not unique; many types of criteria can be used. In most current text classification applications, the main concern is the measurement of classification effectiveness, so we also focus on the classification effectiveness of the obtained classifiers.

Text classification effectiveness can be measured by many performance evaluation indicators, such as Recall, Precision, Accuracy, Error rate, and comprehensive evaluation values of recall and precision such as Eleven-point interpolated average precision and Breakeven point, etc. Since the commonly used criteria is the accuracy, we measure the accuracy of the obtained classifiers. Accuracy is defined as the proportion of samples correctly classified by classifier to all test samples. It is defined as Equation (4.1).

$$acc = \frac{a}{a+b} \times 100\% \tag{4.1}$$

Here 'a' represents the number of input texts which are correctly classified by a classifier, and 'b' represents the number of input texts which are incorrectly classified by a classifier.

4.3 Experimental setting

To evaluate our proposed method, classifiers of the gender classification are trained and evaluated in the following settings.

- The source and target domains are the same. That is, we use Twitter or blog dataset for both the training and test data.
- The source and target domains are different, but classifiers are trained without any domain adaptation techniques.
- The source and target domains are different. The classifiers are trained with a domain adaptation method that considers inconsistency between the source and target domains.

In the third case, three different domain adaptation methods are applied:

- Cut-off method
- Fill-up method
- Fill-up with word similarity method

Note that the third method is our proposed method, while the rest is a baseline.

Table 4.2: Comparison between Naive Bayes and SVM

	Twitter	Blog
Naive Bayes	60.23%	65.48%
SVM	63.45%	70.36%

Table 4.3: Comparison of kernel functions of SVM

	Twitter	Blog
linear function	63.45%	70.36%
polynomial function	57.47%	58.58%
radial basis function	56.79%	62.25%

4.4 Results of gender classification

In this section, results of simple gender classification are reported. Here the difference of the domains in the training and test data is not considered. That is, the source and target domains are the same in this experiment.

4.4.1 Comparison of machine learning algorithm

We compare Naive Bayes and Support Vector Machine(SVM). We use Sklearn[2] which is an efficient library for implementation of Naive Bayes and LIBSVM which is a powerful library for implementation of SVM[3]. The results of two classifiers are shown in Table 4.2. We found that SVM outperformed Naive Bayes on both Twitter and blog datasets. In the succeeding experiments, we always use SVM to train gender classifiers.

4.4.2 Comparison of kernel function

In this subsection, we investigate SVM models with different kernel functions. We compare the linear function, polynomial function and radial basis function. Table 4.3 reveals the accuracy of SVM of three kernel functions on two datasets. It indicates that SVM with the linear kernel function works the best among three kernel functions. It achieves 63% and 70% accuracy on Twitter and blog datasets, respectively.

[2]https://scikit-learn.org

4.4.3 Results of feature selection

Feature selection is performed as explained in Subsection 3.2.3. We train the SVM classifier with the linear function using two different feature sets. One is the top 5% features where the features are sorted by the χ^2 values, the other is the top 10,000 features. Table 4.4 shows the number of features before and after the feature selection. It also shows the number of the common and domain specific feature before and after feature selection.

Table 4.4: Number of features

	Twitter	Blog
total features	1,270,992	6,523,087
5% features with highest χ^2 value	63,549	326,154
common features	17,175	17,175
domain specific features	46,374	308,979
10000 features with highest χ^2 value	10,000	10,000
common features	1,748	1,748
domain specific features	8,252	8,252

Table 4.5 shows the accuracy of the classifiers with two different feature selection methods as well as without feature selection. The feature selection methods slightly decrease the accuracy, but the gap of the accuracy is less than 1%. On the other hand, the time for training the classifiers is significantly reduced due to using the small number of the features. That is, we can use substantially less features to achieve similar performance. In the experiments of the domain adaptation reported in Section 4.5, we always apply feature selection methods for training the classifiers.

Table 4.5: Accuracy of classifiers trained with different feature sets

	Twitter	Blog
All features	63.45%	70.36%
5% features with highest χ^2 value	63.08%	69.75%
10000 features with highest χ^2 value	63.30%	69.72%

4.4.4 Comparsion of domains

Comparing two datasets, the accuracy on the blog dataset was greater than that on Twitter dataset in the results of Table 4.2, Table 4.3 and Table 4.5.

It means that the gender classification for blog articles is easier than for Twitter.

4.5 Result of domain adaptation

Table 4.6 shows the accuracy of several methods of domain adaptation. Two cases are evaluated: one is the case where the source domain is Twitter dataset and the target domain is blog dataset, the other case is vice versa. The cut-off and fill-up are the baselines, while fill-up with word similarity method is our proposed method. We compare these domain adaptation methods with two feature sets: top 5% and 10,000 features selected by χ^2 based feature selection.

Based on the comparison between the cases that source and target domains are (1) the same and (2) different, the case (1) is better than the case (2). This is a perfect example of just what we expressed before, that is, the model trained on one domain often achieves worse performance on another domain.

Comparing the method without domain adaptation and three methods using the domain adaptation, we can see that the accuracy of the model using domain adaptation methods are improved almost 5%. It powerfully corroborates that the domain adaption is effective and important, and it should be paid enough attention to.

Comparing two baselines, the fill-up method is better than the cut-off method in both feature sets. This may be because the fill-up method extends the feature space so that the discriminative information can be considered although it keeps domain specific features weighted as zero.

Finally, we found that our proposed method was better than the baseline methods. It shows that the idea to explicitly add domain specific features that are similar to the features in the sentence is effective for the domain adaptation.

4.6 Error analysis

We conducted an error analysis of Twitter and blog dataset to investigate major causes of errors.

The major problem of the gender classification was the errors of preprocessing of dataset. The details are shown as follows.

- Some non-English words can not be recognized with high possibility since there are so many incorrect spelling and reduplicated writing

Table 4.6: Results of domain adaptation

model		source=Twitter target=Blog	source=Blog target=Twitter
Feature Selection	Domain Adaptation		
no	no	54.53%	53.08%
top 5%	no	54.64%	53.00%
top 5%	cut-off	55.67%	55.34%
top 5%	fill-up	57.73%	58.55%
top 5%	proposed method	59.97%	65.55%
top 10,000	no	53.97%	53.05%
top 10,000	cut-off	55.09%	56.07%
top 10,000	fill-up	57.67%	58.48%
top 10,000	proposed method	59.49%	64.24%

style such as "hfsdghjfgsd", "sooooo".

- Replacement of URL by regular expression did not work perfectly. Especially, some irregular expression about website address causes wrong replacement. For example, when "http://www.ijcis/info." is failed to be recognized as URL, it is divided into "http", "www", "ijcis", "info" by tokenization. In this situation, a large number of useless words are produced by wrong preprocessing.

- Lemmertization is performed as preprocessing. It can convert a word in an inflected form into its base from. However, the conversion of words across parts-of-speech (POSs) may be useful as preprocessing. For example, the adjective "ambigous" and the noun "ambiguity" have different POSs, but these two words represent a similar meaning. Thus the conversion from "ambigous" to "ambiguity" may be useful. However, an ordinary tool for lemmatization does not support such conversion.

The major problem of the proposed domain adaptation method was features newly introduced in a feature vector. Domain specific features are newly added when they are similar to an original feature in a sentence. The similarity between words is measured by word embedding. However, word embedding reflects not only semantic similarity but also syntactic similarity. Therefore, antonyms such as "hot" and "cold" are recognized as similar words by word embedding. Note that these two words are adjectives and similar from a syntactic point of view. However, adding antonym as a new feature is inappropriate for the gender classification.

Chapter 5

Conclusion

5.1 Summary

In this paper, we investigated a new problem setting, which was domain adaptation of gender classification of the Web documents. We proposed a new method for this task. The achievement of this thesis was related to two subtasks: gender classification and domain adaptation.

The first subtask was a kind of classification of texts written in Twitter and blog. The goal of this subtask was to classify a gender of an author of an English text into female or male. We investigated features for machine learning, feature selection methods and training algorithms. The word unigram and bi-gram were extracted as features with a few preprocessing. As for the feature selection, χ^2 values were used to choose the most 5% or 10,000 effective features. Finally, we used Support Vector Machine (SVM) and Naive Bayes model as supervised machine learning algorithms.

The other subtask was the domain adaptation based on feature expansion of similar words. Our proposed method consisted of several steps. First, using word embedding pre-trained from a huge collection of texts, we calculated the similarity between a feature (word) in a sentence and another domain specific feature (word) that did not appear in the sentence. Second, we chose several features with high similarity. Finally, we explicitly added the selected features into the feature vector by reweighting them as 1, keeping the common features at the same time. Our proposed method was compared with two baseline methods. One was the cut-off method that used only common features. The other was the fill-up method that used all common and domain specific features where the weights of specific features of the other domain were set as 0.

The results of our experiments showed that domain adaptation was im-

portant since we confirmed the difference of the source and target domains caused a significant drop of the accuracy of the gender classification. All domain adaptation methods improved the accuracy. The classifier trained by the cut-off method had the better performance than one without domain adaptation with the improvement of 1% to 3%. The fill-up method had the better performance with the improvement of 3% to 5%. Finally, our proposed method had the better performance than these two baselines with the improvement of 5% to 10%.

5.2 Future Work

Although the proposed method of the domain adaptation outperformed the baseline methods, there is still room to improve the accuracy of the gender classification on different domains. Future work of this study is summarized as follows.

- To investigate an appropriate method to delete non-English words.
- To design more sophisticated regular expression for pattern matching to recognize URL.
- To normalize word forms accross different POSs.
- To improve the method to calculate word similarity.
- To investigate a new and preferable domain adaptation method.

Bibliography

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606, 2016.

[2] John D. Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating gender on Twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1301–1309, 2011.

[3] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[4] Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *Advances in neural information processing systems*, pages 2456–2464, 2011.

[5] Yao-Tsung Chen and Meng Chang Chen. Using chi-square statistics to measure similarities for text categorization. *Expert systems with applications*, 38(4):3085–3090, 2011.

[6] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126, 2006.

[7] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142, 1998.

[8] Wouter M. Kouw, Laurens J.P. Van Der Maaten, Jesse H. Krijthe, and Marco Loog. Feature-level domain adaptation. *The Journal of Machine Learning Research*, 17(1):5943–5974, 2016.

[9] Muhammad Mansur and Britton Wolfe. Gender classification of mobile application reviews. https://www.semanticscholar.org/paper/Gender-Classification-of-Mobile-Application-Reviews-Mansur/1aff9ae0472ea71f904968e1cfee7f912edcf591, 2014.

[10] Paul Michel and Graham Neubig. Mtnt: A testbed for machine translation of noisy text. *arXiv preprint arXiv:1809.00388*, 2018.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[12] Arjun Mukherjee and Bing Liu. Improving gender classification of blog authors. In *Proceedings of the 2010 conference on Empirical Methods in natural Language Processing*, pages 207–217, 2010.

[13] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[14] Wei Pengfei, Ke Yiping, and Goh Chi Keong. Domain specific feature transfer for hybrid domain adaptation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1027–1032, 2017.

[15] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[16] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

[17] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, 2014.

[18] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[19] Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria. Feature ensemble plus sample selection: domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3):10–18, 2013.

[20] Xiang Yan and Ling Yan. Gender classification of weblog authors. In *AAAI spring symposium: computational approaches to analyzing weblogs*, pages 228–230, 2006.

[21] Zi Yin and Yuanyuan Shen. On the dimensionality of word embedding. In *Advances in Neural Information Processing Systems*, pages 887–898, 2018.