Spring 2012

CROSS-LINGUAL TEXT CLASSIFICATION WITH MODEL TRANSLATION AND DOCUMENT TRANSLATION

Zhang Zhang
San Jose State University

CROSS-LINGUAL TEXT CLASSIFICATION WITH MODEL TRANSLATION AND DOCUMENT TRANSLATION

A Thesis

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Zhang Zhang

Jan. 2012

The Designated Thesis Committee Approves the Thesis Titled

CROSS-LINGUAL TEXT CLASSIFICATION WITH MODEL TRANSLATION AND DOCUMENT TRANSLATION

by

Zhang Zhang

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

Jan. 2012

Dr. Teng Sheng Moh	Department of Computer Science
Dr. Mark Stamp	Department of Computer Science
Dr. Robert Chun	Department of Computer Science

ABSTRACT

CROSS-LINGUAL TEXT CLASSIFICATION WITH MODEL TRANSLATION AND DOCUMENT TRANSLATION

by Zhang Zhang

Most enterprise search engines employ data mining classifiers to classify documents. Along with the economic globalization, many companies are starting to have overseas branches or divisions. Those branches are using local languages in documents and emails. When a classifier tries to categorize those documents in another language, the trained model in mono-lingual will not work. The most direct solution would be to translate those documents in other languages into one language by the machine translator. But this solution suffers from inaccuracy of the machine translation, and the over-head work is economically inefficient. Another approach is to translate the feature extracted from one language to another language and use them to classify another language. This approach is efficient but faces a translation inaccuracy and language culture gap. In this project, the author proposes a new method which adapts both the model translation and document translation. This method can take advantage of the very best functionality between both the document translation and model translation methods.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

Most enterprise search engines employ data mining classifiers to classify documents. Spurred by economic globalization, many companies now are starting to have overseas branches and divisions, and these branchs are utilizing localized language in documents and emails. When search engines try to categorize those documents in other languages, the trained model in mono-language will not work. This presents a new problem called Cross-Lingual Text Classification(CLTC) to the search engine.

1.1 Cross-Lingual Text Categorization (CLTC)

Cross-Lingual Text Categorization is a relatively new topic, the earliest known paper is in 2003 by Nuria [BKV03]. In this initial paper, the problem of Cross-Lingual Text Categorization (CLTC) was first introduced. The author Nuria suggests to the reader that we learn the lesson from an already known area called Cross-Lingual Information Retrieval (CLIR), and the author goes on to compare the difference between CLTC and CLIR. From the lesson learned from CLIR, there are three basic approaches to resolve a CLIR problem, translate document, translate query and create language independent model. For CLTC, we also have three basic approaches which include 1) translation documents, 2) translation model and 3) creation of a language independent model. Also, the author points out that in CLTC there are two cases, 1) poly-lingual training and 2) cross-lingual training. The difference between the poly-lingual and cross-lingual is for poly-lingual training, we have labeled data for all languages, but for cross-lingual training, we only have labeled corpus for one base line language. This paper will focus on cross-lingual training which equates to labeled data only available in one language.

Because of the lack of standard documents, the author only used a limited number of documents and only two western languages, namely Spanish and English.

1.2 Existing Solutions

Most current direct solutions utilize machine translation software to translate documents into mono-language, and then train the classifier, but this approach has some concerns and issues. Machine translation software based on translation probability can not translate documents with 100% accuracy. Even thought the state of art machine translation softwares in the industry like Google Translator and Microsoft Translator, sometimes the translation does not make sense. A lot of

overhead work is required to be done, for example, corpus need to be segmented and then sent to APIs of those translators chunk by chunk. From an efficiency point of view, this way is neither time nor economically efficient. Every time you call out a translation API to translate a document, it will takes an average of 1 second for the translation part to finish the job, and if we have 10,000 documents, it will takes over 2.7 hours to translate all of them. Google Translator API has recently started to charge user $20 for every million words and no free API after July 2011. Microsoft Translator still has a free API but it limits the transactions to 200 per day per user, and developers have to pay after 200 translations based on the total transaction amount. In real enterprise search engine scenarios, because of security concerns, most of the companies do not allow exposing their internal data through HTTP. Moreover they would prefer to buy or rent a dedicated translate server inside their own intranet domain, which will cost alot more money depending on the vendor.

The second approach is to translate the feature extracted from labeled corpus, the feature selection algorithm selects a subset of words used in the corpus as features, those features been selected are translated to different languages, then a new model is trained base on those translated features for each language. The feature extraction algorithms pick only those features that in the scope of target languages. This approach is efficient from a time and cost perspective, because it does not require a machine translation. It only needs a bilingual dictionary on local file system to translate the selected features, but it can suffer from the inaccuracy of dictionary translation, in that words may have different meanings in different context. So how to select the translation as features is tricky.

Another approach is to train a language independent model. This approach avoids the need for translation on both documents and features, and the trained model can work with any language, but it has the lowest accuracy among the three approaches mentioned. Language independent method is trying to summarize common features or characteristics between languages, this method has to drop language or culture specific features and characteristics, this is the reason why it has lowest accuracy, because the model only need to be trained once, thus it is saves effort of translation and re-train the model.

From the classification accuracy point of view, the document translation method normally has the highest accuracy for the same data, whereas the model translation method has the second highest accuracy, and independent model method has the lowest accuracy

1.3 Proposed Solution

In this paper, I propose a new approach which incorporates the best parts from both the document translation and model translation methods in to one cohesive method. My approach favors the ability to translate part of the test documents, but not all of them, thus saving time on the effort of translating by

machine translator. And the key finding is: Utilizing the feature extraction algorithms and combining algorithms on translated test documents as well as training documents, enhances the model translation method with better accuracy than before.

1.4 Organization of the Paper

In this paper, Chapter 1 introduces the definition of Cross-Lingual Text Classification. Chapter 2 reviews previous works related with CLTC. Chapter 3 proposes our new approach to the problem of CLTC. Chapter 4 states how the testing documents in different languages was gathered and pre-processed before being used to train and classify documents. Chapter 5 explain the way that we test and verify our new approach with Support Vector Machines and Naïve Bayes classifiers. Chapter 6 is the conclusion and future works.

CHAPTER 2

RELATED WORKS

Since translation accuracy is the big factor in both model translation and document translation. B.M.Amine [AM07] in 2007 use WordNet® [Mil95] [Fel98] to improve the translation accuracy and generate conceptual category profiles in the training phase. In the classification phase, author used machine translation to translate unlabeled documents into English and generate conceptual vector with aid of WordNet®. WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Author uses this feature of WordNet® to blur the gap of translation, since the WordNet® not avaliable in all languages, it is still constricted by the cultural differences. Instead of using standard classifiers, the author choices to make the classifier himself base on conceptual vector by calculating distances between weighted vectors. From the test result data, the algorithm used to evaluate the weight and calculate distances still needing improvements.

Shi and other two researcher from Yahoo research lab also published a paper [SMT10]. In this paper, the author focuses on improving translation accuracy by learning the translation probabilities with an EM algorithm, and the translation is limited to the scope of context. The observational finding is that in the typical translation process, the feature would be translated by dictionary, however, this translation method has major drawbacks from ambiguity of transition, and inaccurate translation will distort the classification accuracy. The translation should rely on the context of the documents. The idea is to limit the translation scope within the target language documents, and locate the most related translation from the documents instead of using frequency based WordNet or direct translation like Machine Translator. The second step of this paper is based on another observation that because of cultural language differences, some words have indirect translation to the source language. So the solution is to hereby re-train the model by semi-supervised algorithms and learn the keywords by itself. The problem is that every time it learns a word, it requires a re-train, and when the training set is large, this method could take a very long time to achieve good classification results.

P. Prettenhofer in 2010 uses the structural correspondence learning [PS10] method to learn the correspondences among the words from both languages by means of a small number of so-called *pivots*, which is a pair of words from source language S and the target language T. This paper has one very strong assumption that the word translation map is always a one to one map, which in real scenarios is not always the truth.

Microsoft and Tencent researchers discovered a new method [NSHC11] by

explorer Wikipedia. Wikipedia, which launched in 2001 has become a very big multilingual and Web-based encyclopedia. The author uses the multilingual content and the links in each document to align the topic representations. The benefit of this approach is that it does not require an external oracle to translate anything, however the drawback is if the categories of document are out of Wikipedias scope, then it cant categorize.

In the above papers which using model translation as the major methodology, the translation accuracy of features is the biggest factor that affect the classification accuracy, also all of those papers are trying to improve the translation accuracy either by using special algorithms or using third party knowledge bases. The accuracy of translation has been improved but not close to the document translation classification result.

Co-Training in data mining has become more and more popular, as it provides a shorter learning curve with better classification results. Wan in 2009 [Wan09] adopted the Co-Training methodology in classification of users review in different languages. It utilized two typical classifiers, SVM and NB. For the experiment, the author used two sets of labeled data, one is in Chinese and the other one is in English. Initially the author will translate the labeled English review to Chinese., He then combines with Chinese reviews, and trains the data model. Then he translates the labeled Chinese review to English. The translator they used is Google Translator, and they achieved 70-80% classification accuracy. The downside of this approach is that it still needs a labeled data set in another language to boost the classification, which in may cases are not available. In this paper, the document translation method was partially used to improve the classification accuracy, documents need to be translated and labeled, which cause efforts on either human translation or machine translation, it is not a effective approach neither timely nor economically.

Ling [LXD+08] in 2008 proposed a language independent method of classification called: *information bottleneck* (IB). The IB method is a distributional learning algorithm, and the clustering and classification problems can be treated as a coding process. The IB technique is used to mine the common part of the pages in different languages for classification. The author compared the IB method with traditional classifiers like Support Vector Machines (SVM), Naïve Bayes Classifier (NBC) and Transductive Support Vector Machines (TSVM), and the results show that IB has only marginal benefit then other classifiers in this area.

CHAPTER 3

PROPOSED APPROACH

3.1 Common Process of Classification

The most commonly used classifiers in enterprise search engines are supervised learning classifiers, like SVM, Naïve Bayes. This type of classifier requires a set of labeled data to train the model and then apply the trained model to unlabeled data. For text classification, Figure 3.1 shows how the classifier fit in the common process of text classification in search engines.

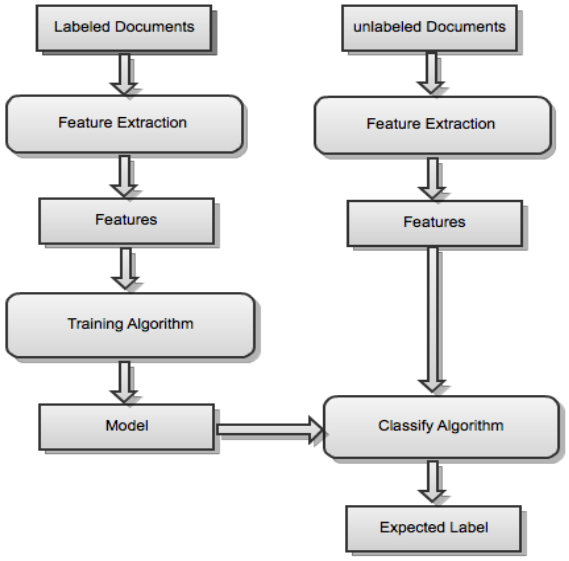

Figure 3.1: Common Process of Text Classification

In Figure 3.1, feature extraction algorithms are applied on labeled corpus. With bag-of-words theory [bag], features will be a sub set of words that are used in corpus with weights, extracted features with label are feed to training algorithms, and training algorithms generate a model based on the training corpus. Unlabeled Corpus will also go through the same algorithm of feature extraction. Then, based on the trained model, and feature from the unlabeled corpus, the classifier gives the classification results.

For the document translation method, the documents in other languages are translated into one language, then use the trained classifier to classify the translated documents, shown in Figure 3.2.

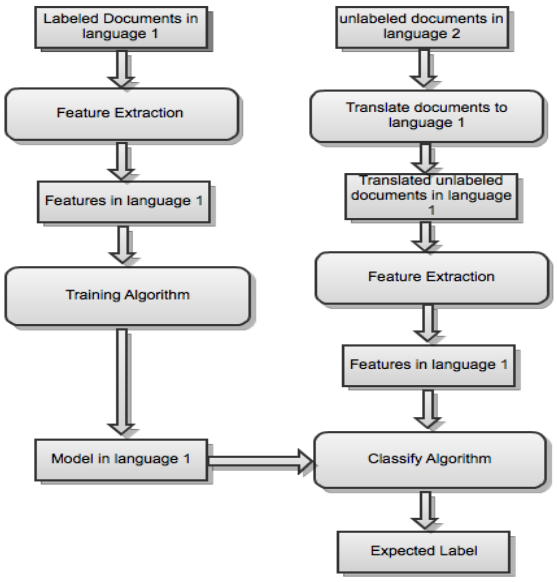

Figure 3.2: Document Translation Method of CLTC

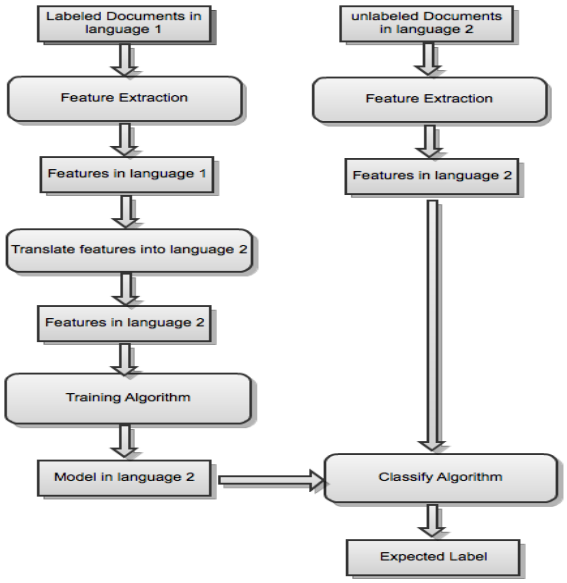

Figure 3.3: Model Translation Method of CLTC

Table 3.1: Number of different features for each category and language

	Es	Fr	Ch
Business	183	278	305
Entertainment	190	287	323
Sports	273	302	366
Health	189	251	313

The model translation method will translate the features into other languages, and then train different classifiers in each languages, shown in Figure 3.3

From this common process, we can discover that the feature extract algorithm is the most language specified part of this process, especially for CJK languages. Also, once the model has been trained from the corpus of one language, it is hard to migrate to another language.

3.2 Overview of Proposed Approach

The benefit of the documentation translation method is in the accuracy. Modern machine translation technology can translate documents into all kinds of different languages with an acceptable accuracy, but the problem is when the number of documents in other languages is large, the API calls to those machine translation engines and the cost for subscribing the APIs are remarkably high. The Model Translation method on the other hand will only require a bilingual dictionary, which is efficient but the translation result is not accurate.

The proposed method is inspired by the observation that the inaccuracy of model translation method mainly because the translated model is missing lots of features that are special to the language and culture. For example, in 3.1, I listed the number of the features that can be found in the language translation method but can not be found in model translation method. I looked into those features, and found that the majority of them are related to culture difference, so we need a way to wisely incorporate those features in the model translation method.

Another observation is that after using the document translation method, we obtained labeled documents. Even though the label is not 100% accurate, we can still take advantage from the translated documents to complete the feature set for the model translation method.

By combining both the model translation and document translation method, translation of all documents into another language wouldnt be required, thus saving the time effort and cost of API calls. After completing the feature set of model translation, the accuracy of model translation will be improved.

The approach has a few steps:

(1) Train Classifier base on English (or the language that majority used).

(2) Randomly select and translate a small partial subset of documents in another language using machine translation software, then apply the model learned in English to classify those documents.

(3) Take the result of classification, find out which class each of the documents belongs to. Extract features out of labeled documents that derive out from the previous step.

(4) Mix the feature extracted from the English training data set and the classified documents in other languages into one big feature set.

(5) Translate the feature set into the target language, then use the translated feature to train a model and then to classify documents in the target language.

From the step (2), there is a percentage of documents in other languages which have been classified correctly with the features carried over from the English documents. Inside those documents in a different language, there must be additional features that can represent the class of the document in that particular language and culture. Those feature will be captured by step (3) with feature the extraction method. Finally, those features will be used to classify other documents. Figure 3.4 shown how the proposed method works.

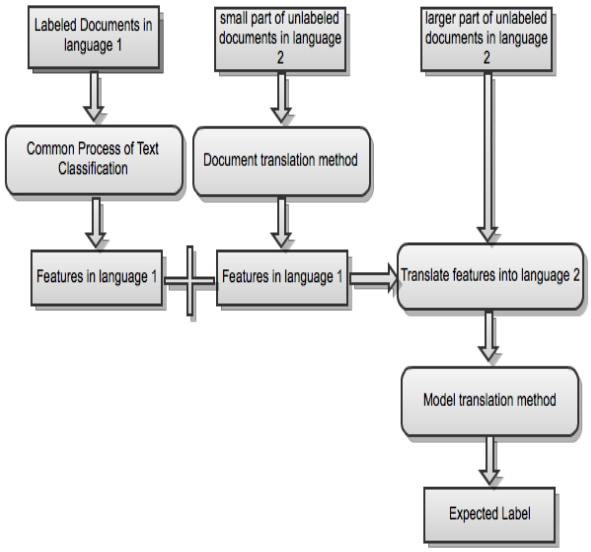

Figure 3.4: Proposed Solution of CLTC

3.3 Feature Selection Algorithm

Feature selection is a technique to select a subset of the features available for describing the data before applying a learning algorithm. It has been widely observed that feature selection can be a powerful tool for simplifying or speeding up computations, and when employed appropriately, it can lead to minimal loss in classification quality. Normally a feature will be associated with a weight value. From my experience, the completion and accuracy of feature select contributes more in the final classification result than in the weighted value associated with the feature.

In previous works, TF (term frequency), TF-IDF (term frequency inverse document frequency) are the most common used feature value. Those values are represent the importance of a feature from statistics perspective.

With bag of words theory, TF-IDF is defined as:

$$tf\text{-}idf(t,d) = tf(t,d) * idf(t)$$

$$idf(t) = log\frac{|D|}{|\{d : t \in d\}|}$$

$|D|$ is the total number of documents in the corpus.
$|\{d : t \in d\}|$ is number of documents where the term t appears.

Threshold, Chi-squared (χ^2) statistic) and IG (Information Gain) are the commonly used feature selection algorithms. Those feature selection algorithms from different perspectives to evaluate each feature and assign a value to represent the importance of the feature, after sort all the features with the importance value, a limited number of the most important features are selected. Reduced number of features will lower the dimension of training classifiers, thus save the time when training, further more, feature selection algorithm can also counteract the negative impact of noisy features.

Threshold is using a human picked fixed feature value as threshold value, then keep all features that value are greater then the threshold, this method is efficient, easy to implement, but the threshold value is different from one dataset to another, so every time when change dataset, the threshold value has to be changed.

3.3.1 Chi-squared

The idea behind the Chi-squared algorithm is comes from a commonly used way of mathematical statistics. The basic idea is to calculate the difference between observed read value and theoretical value to decide if the theory is correct or not. We assumed that the two variables are independent, and then observe how much the difference there is between the observed value and the theory value. If the difference

is small enough, we will consider that the error is a natural sample error, it happens because the method we use to measure the value is not accurate or accidentally, the original assumption is true. If the error is so big that there is no way it happens because of measure or accident, then the original assumption is false.

We use E to represent the theory value, observed value is x, then the difference can be calculated in following equation.

$$\sum_{i=1}^{n} \frac{(x_i - E)^2}{E}$$

When the difference is calculated, the difference is compared with a threshold value, and if the difference is greater then the threshold value, then the original assumption is false, otherwise the original assumption is true.

Lets apply this theory to feature selection. Assume that we have N documents, M of then are related with sports, we want to see how the word "football" and category "sports" are related, we have four values that can be used:

(1) Number of documents that include football and belongs to sports, denote as A.

(2) Number of documents that include football but not belongs to sports, denote as B.

(3) Number of documents that not include football but belongs to sports, denote as C.

(4) Number of documents that not include football and not belongs to sports, denote as D.

We use a Table 3.2 to make it more clear. A+C actually means all documents in sports category, so it is equal to M, also B+D=M-N.

Now if the assumption that "football" has nothing related with "sports", then for all the documents, the probability of "football" present should be equal for all category. We do not know what is the value of this probability, but we should be able to get the value from observation. This value should be close to:

$$\frac{A+B}{N}$$

The number of documents in "sports" cateogry is A+C, so in those documents should have following number of documents include "football".

$$E_{11} = (A+C)\frac{A+B}{N}$$

Which is the theory value, put this value in equation of calculate difference value

Table 3.2: Chi-squared Statistic Denote

Feature Select Including Feature F	Belongs to Category X A	Not Belongs to Category X B	Total A+B
Not Including Feature F	C	D	C+D
Total	A+C	B+D	N

$$D_{11} = \frac{(A - E_{11})^2}{E_{11}}$$

With the same way we can calculate D_{12}, D_{21}, D_{22}

$$\chi^2(\text{"}football\text{"}, \text{"}sport\text{"}) = D_{11} + D_{12} + D_{21} + D_{22}$$

more generally.

$$\chi^2(t, c) = \frac{N(AD - BC)^2}{(A + C)(A + B)(B + D)(C + D)}$$

Since N, (A+C) and (B+D) are same for all words in same category, and when applying feature selection algorithm, we only concern about the sorted sequence rather then the real value of Chi-squared. So the Chi-squared can be simplified to:

$$\chi^2(t, c) = \frac{(AD - BC)^2}{(A + B)(C + D)}$$

The simplified version of χ^2 is used for selecting features from documents.

3.3.2 Information Gain

Besides Chi-squared, Information Gain (IG) is also a very effective method. All feature selection algorithm, always quantization the importance of feature then select, how to quantize the importance of one feature, makes the difference of algorithms. In Chi-squared, the relevance between the feature and the category is quantized, the more relevant the higher of feature value, the feature should be preserved.

In Information Gain, the importance of feature is quantized by how much information of the feature brings to the system, important feature brings more information.

From Shannon theory, the entropy of information can be calculated with following:

$$H(X) = -\sum_{i=1}^{n} P_i log_2 P_i$$

For a categorization problem, category C is a variable, which can be one of C_1, C_2, ..., C_n, for each value of C, the probability is $P(C_1)$, $P(C_2)$, ..., $P(C_n)$, n is the number of categories. The entropy of this categorization system can be calculated with:

$$H(C) = -\sum_{i=1}^{n} C_i log_2 C_i$$

Information Gain is calculated for each feature. Look into the fact that for a feature t, what is the different of the entropy between has feature t and does not have feature t, the value of difference is how much entropy the feature brings to the system, which is also known as Information Gain.

When the system doesn't include feature t, we can derive that feature t exists, but it is in constant value and cannot be changed, hance it will not bring information to the system. The entropy of this scenario called conditional entropy, the condition is the t is fixed, denote as $H(C|X = x_i)$. We need to calculate conditional entropy for all possible value of x_i:

$$H(C|X) = \sum_{i=1}^{n} P_i H(C|X = x_i)$$

Since for text categorization, feature t only has two possible value, either t present (t) or not present ($\bar{t}$):

$$H(C|T) = P(t)H(C|t) + P(\bar{t})H(C|\bar{t})$$

Thus, IG value can be calculated with:

$$IG(T) = H(C) - H(C|T) = -\sum_{i=1}^{n} P(C_i) log_2 P(C_i) +$$

$$P(t)\sum_{i=1}^{n} P(C_i|t) log_2 P(C_i|t) + P(\bar{t})\sum_{i=1}^{n} P(C_i|\bar{t}) log_2 P(C_i|\bar{t})$$

$P(C_i)$ denote the probability of category C_i, $P(t)$ denote the probability of feature t present, can be calculated with number of documents that has t divide by total number of documents, $P(C_i|t)$ denote when t present, the probability of category C_i.

Information Gain considers both the feature present and not present, same as Chi-squared, so it has satisfied results for the classification problem. The difference between IG and Chi-squared is that Chi-squared consider how well the feature

Table 3.3: Feature Selection Method Compare

TFIDF	Chi-squared	IG
92.80%	96.49%	88.80%
91.80%	94.12%	82.92%
93.20%	95.97%	88.50%
94.80%	97.38%	89.40%

contribute to the system, rather then individual category, so it makes Chi-squared good for global feature selections, but not local feature selections.

I implemented both Chi-squared and IG algorithm, select part of the English document to compared the categorization result with different feature selection algorithms, and the result is shown in following Table 3.3.

In this Table 3.3, TFIDF method is using TFIDF as the feature selection value and sort descant. In all three method, a limit number of features are selected, 1000 for each category, when the feature is cross categories, the one with biggest weight will be selected for that category.

From the test result of different feature selection algorithms, also referenced [DDH$^+$07] and [LXZH09], I selected Chi-squared to select feature and use TF-IDF value as the weight which gives me best classification result.

CHAPTER 4

DATA SET

4.1 Data Statistic

There is no standard test data in this field yet, so I started a scheduled task in amazon cloud and crawling rss news feed from Yahoo and Google news. The crawler was started on June 2011 and at the end of October 2011. I have gathered over 40,000 documents in 4 different languages, including English, Spanish, French and Chinese. Non-english documents will be translated into English by using Microsoft translator for classification result compare purpose. Following Table 4.1 are the distribution of documents in languages, all documents are in 4 categories: business, entertainment, sports and health. The reason why I chose these 4 categories is because they overlap and the data sources are available across all 4 languages.

4.2 Data Storage

Solr [Fun06] is a popular, fast open source enterprise search platform which was written in pure Java from the Apache Lucene project. The major features include powerful full-text search, hit highlighting, faceted search, and dynamic clustering. It is highly scalable, providing distributed search and index replication, and it powers the search and navigation features of many of the worlds largest internet sites. It is used for this project to create the search index, providing data source for classification. The Solr schema includes fields like unique id, language, title, content, translation and category. Solr instance is hosted in the Amazon Cloud machine with a static IP address, since Solr runs within a servlet container, crawler machines and classification machines use HTTP protocol to communicate with the search index. Solr has its own search query language which will be used to select data sets from the index.

Table 4.1: Data Statistic

	English	Spanish	Chinese	French
Business	4869	1926	1713	2811
Entertainment	5987	785	1253	4841
Health	2873	831	1598	1224
Sports	2739	2191	2214	2412

4.3 Crawlers

Figure 4.1 shows the data crawling and storage design. Crawlers respond for gathering data in a periodical manner and transform the data into the format that can be parsed by Solr indexer, crawlers will also send the data to Solr indexer. The way I gather data is first, I manually pick the RSS feed source from Google news and Yahoo news for those 4 categories. I initially write a Java program to parse the RSS XML, then go to the URL which RSS item is pointing, and get the HTML source. Due to the format of the HTML, it can be very different from one website to another and from one language to another, so I use a clustering algorithm to cluster all the text paragraph and pick the largest cluster as the content of news. Because the clustering data is out of the scope of this project goal and if this algorithm failed to find content, then the item of RSS will be dropped and make no impact on classification result, I didnt do further analysis on this algorithm. Finally, the crawler will transform the data it gathered. If the data is not in English, it will call the Microsoft translator API to translate the content into English, Solr index accept XML format as data source, all HTML tags will be trimmed before put into the XML, so for each field of the Solr data XML, it will be plan text. Crawler is scheduled as cron tasks in Linux and will run multiple times a day, URL of the news item is the uniqueness key for each news, if the URL is duplicate, Solr will overwrite the previous data.

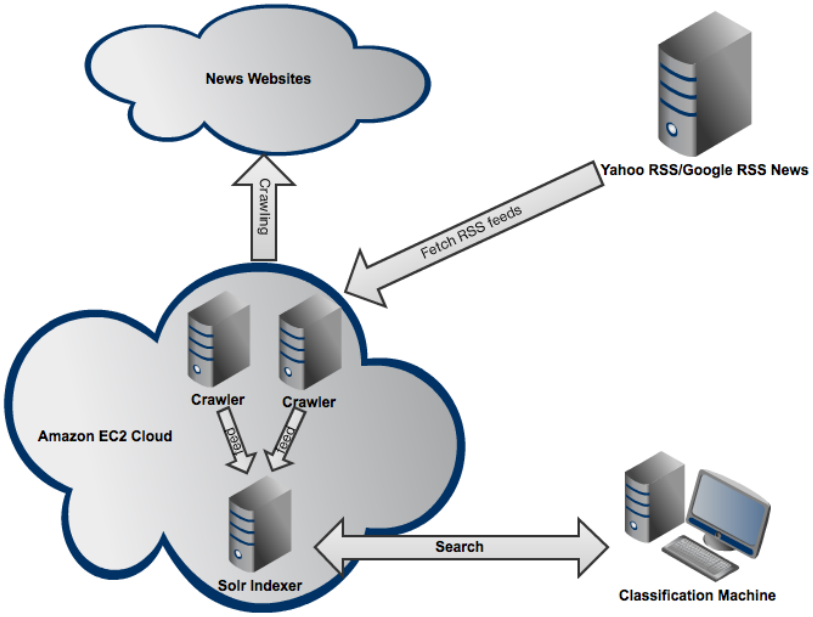

Figure 4.1: Data Storage and Crawling

4.4 Data Pre-processing

After the Solr indexer gets the data feed from the Crawler, all unnecessary spaces and line breaks are trimmed. Next, the data will be stored to the index without tokenizing the filter and stemming at this point. When the classification machine fetched the data from the Solr index, following preprocess is applied to the data. English, Spanish and French documents are tokenized with space and punctuation marks, Chinese documents are not tokenized. Second, the tokenized string will go through a stop word filter for each language, Chinese documents are not put through this process. I attempted to do stemming on the token but because of leaking stemming algorithms for Spanish and French, non-stemming actually gives a better result for English, so I decide not to do stemming.

4.5 CJK Special Processing

Because the Chinese, Japanese and Korean (CJK) languages does not have a natural splitter for words like latin languages, when processing CJK languages, tokenizing becomes very difficult. The longest Matching Word is a common algorithm to tokenizing CJK languages, and it will search a dictionary for the longest word that matches the word segment to tokenize. I tried this algorithm, but it does not give a satisfactory result because of the large gap between the translated feature and the segment word features. Further more, it has a negative impact of the classification result. For the CLTC task, tokenizing is not absolutely necessary. If one is choosing a document translation, then the problem becomes a tokenized problem of English. If one is choosing a model translation, matching the word directly through the CJK document also can tell you if the feature word is in the document or not. The one issue is that, since the total number of tokenized words are not available, calculating the TF-IDF value is impossible.

From this observation of classification results with different feature selection algorithms, the weight of the feature is not the major contributor of the classification result. Rather, the feature selection algorithm is the major contributor. From my experience, adjusting the weight of feature or choosing the different algorithm to calculate the weight can only improve the classification result by 1-2%. Contrast this with the feature selection algorithm which can improve the classification result by 5-8% for the same document base.

So I decided to give an arbitrary value for the weighted value, and this method gives me a fair enough result and much better than tokenizing does. See the results shown in Table 4.2.

Table 4.2: Chinese tokenizing vs non tokenizing

	English	zh_CN With tokenize	zh_CN weight = 1
Business	96.49%	50.00%	88.75%
Entertainment	94.12%	50.62%	91.36%
Health	95.97%	50.62%	83.95%
Sports	97.38%	51.25%	78.75%

CHAPTER 5

EXPERIMENT AND RESULT

5.1 Overview of the Experiment

With the data set I gathered, documents in English are used as the base language. Spanish, French and Chinese documents are used as documents in other languages. First, a mono-lingual classifier is trained with English documents, and tested with English documents, then the same classifier is used to classify the unlabeled English documents that translated from Spanish, French and Chinese, this is the document translation method. Second, the features used in training the classifier are translated into Spanish, French and Chinese, a new classifier is trained for each language, and used to classify the unlabeled documents in original Spanish, French and Chinese, this is the model translation method. The classification result from the document translation method and the model translation method are saved in files, so we can use them to later compare with the classification result from proposed method. Using proposed method, small randomly distributed part of the labeled documents from the classification results of the document translation method in three different languages are selected, feature selection algorithm is used with those selected documents in their original languages to extract the features. The extracted features are saved and combined with the features from the model translation method. Then a new classifier is trained for each language, base on the combined features set. Those classifiers are trying to classify the very same documents that used in document translation method and the model translation method, classification result is recored.

In order to demonstrate the proposed approach of combining the document translation method with the model translation method is improves the accuracy of cross-lingual classification and it is not dependent on specific classifiers, I used both Support Vector Machines (SVM) and Naïve Bayes classifiers.

5.2 Support Vector Machine

The SVM (Support Vector Machine) classifier [Vap95] was first introduced by Vapnik, Vladimir N. in 1995. It provides special advantages in the area of small sample and high dimensional classification.

But due to the high computation effort, the usage was limited until 1999, Joachims publish a method ([TJBe99]) that makes the large scale learning with SVM become practical.

In this paper, I used SVM^{light} as the experiment SVM classifier, SVM^{light} is an implementation of Vapnik's Support Vector Machine for the problem of pattern

recognition, for the problem of regression, and for the problem of learning a ranking function. The optimization algorithms used in SVM^{light} are described in [TJBe99]. The algorithm has scalable memory requirements and can handle problems with many thousands of support vectors efficiently.

5.2.1 Parameter of SVM^{light}

SVM^{light} support different kernels and lots of different parameters to tweak the performance and behavior, I compared the different kernels and parameters and find out that for this text classification problem, linear kernels with default parameters give the very best classification result.

5.2.2 Wrapper of SVM^{light}

Since all my project is base on the Java platform, the SVM^{light} is written in C/C++, so I wrote a wrapper and use it to integrate SVM^{light} with Java. The wrapper responses for calling the SVM^{light} executable in a disk folder with correct parameters, and captures the output of SVM^{light}, then store it to a file on disk.

5.2.3 Data Splitting

SVM is a liner classifier which attempts to maximum the margin between two hyper spaces, thus it is a two class classifier. We have four classes, with one vs all method, I used the classify scheme showed in Figure 5.1 to organize documents. One document from each class and use SVM to separate one from the other three, but I was facing the problem that due to the data being unbalanced, the classification result is biased to the larger number of training data.

Because of this, I used the method to select only one document from each of the other 3 classes, so the classifier will have balanced data to learn. Which is shown in Figure 5.2.

5.2.4 Performance

Since there are over 40,000 documents, each document in average has a length of 200 words, and that means there are over 8 million words needed to be analyzed. There is also a more important need to calculate the statistic values for each words, like TFIDF, Chi-squared Statistic. Without performance tweaks and improvements, the task wouldnt finish in hours or even in days. After analysis of the program, the TFIDF calculation was identified as the most compute sensitive part.

In order to improve the performance, Java multi-threading is used, based on the number of CPU, the thread pool with twice the size of the CPU number is created to calculate the TFIDF value. All of the threads are dispatched with task of one document. After this is finished, it will take another one from the pool until all

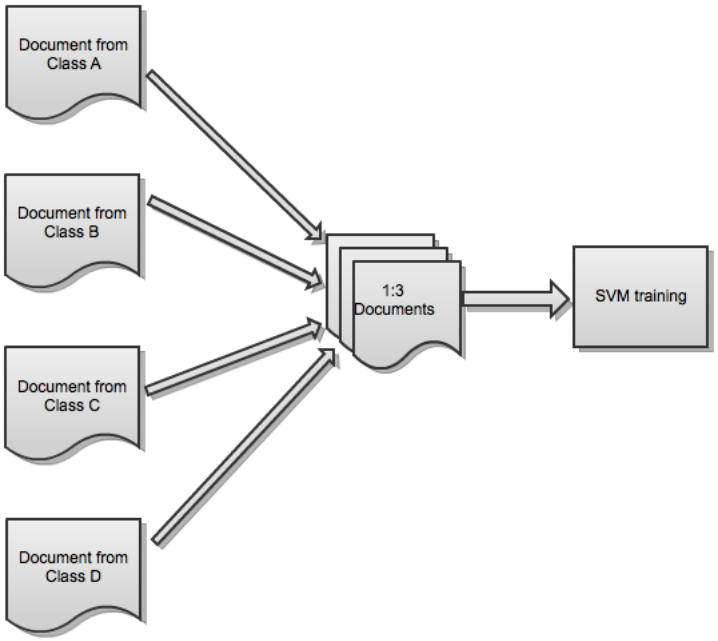

Figure 5.1: Data Split Design which suffer from data unbalancing

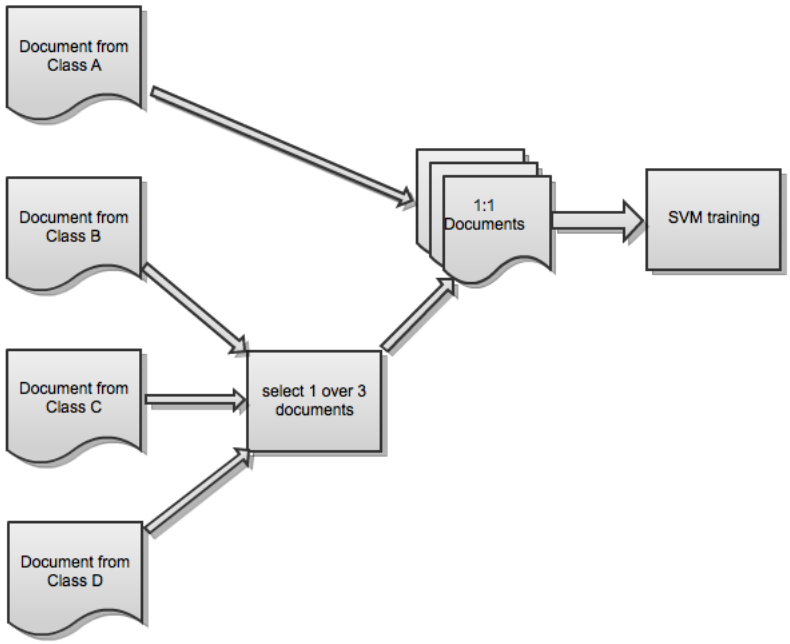

Figure 5.2: Data Split Design with balanced distribution

Table 5.1: Base Line Classify Results

	En	Es_doc	Fr_doc	Ch_doc	Es_mdl	Fr_mdl	Ch_mdl
Business	96.49%	84.17%	89.02%	92.17%	79.99%	79.67%	84.83%
Entertainment	94.12%	83.02%	87.37%	91.17%	71.65%	78.77%	92.33%
Health	95.97%	89.32%	82.58%	82.77%	52.68%	83.05%	76.77%
Sports	97.38%	89.02%	92.57%	94.47%	90.23%	91.36%	82.73%

documents have been calculated. With this multi-threading method, the time to compute all documents has dropped down to just a couple of hours.

5.2.5 Base Line

The goal of this approach is to improve the classification accuracy of CLTC, the existing approach with document translation and model translation becomes the base line. The results of my approach are then be compared with those base line classification results.

The documents translation method will use the documents which are stored in the Solr index and already translated by Google Translator©. With the model translation method, the features are also translated by Google Translator©, but the translation result is stored in local text file and the local text file is act as a bilingual dictionary, meaning that for same words it will only call up the translator ones.

Base Line test is using SVM^{light}, English Documents are split into two parts, the training part and the test part. Each part has half of the documents for all four categories. All documents in Chinese, Spanish and French are half size of English documents used for the test part for the model. A English model is trained with the training data set from English documents, the model is first verified with test documents in English, then apply the same model to the document translation method for all of the other three languages. After translating the feature words in the English model to the other three languages with dictionary, the model translation method is applied to all three languages. The base line is using accuracy in percentage, the higher the better. Base line classify results are shown in Table 5.1.

In Table 5.1, Es denotes Spanish, Fr denotes French, Ch denotes Chinese and En denotes English. Suffix _doc means it is using the document translation method, _mdl means it is using the model translation method.

From the Base line test results, we learn that the English test documents has bast performance over all other languages, for the other three languages, the document translation method has a higher accuracy than the model translation.

5.2.6 Learning Stage

From the document translation classification results, there must be some information we can take advantage of. In the base line test, all documents in Non English categories are used, and now documents in the Non English are splitted into two parts. The first part is the training set which uses the model translation method to train the model. The second part is the test set, which used to verify the model classification result.

English training documents are used to train an English model, and then this model is applied to translated documents in the Non English training set. The training set is labeled with the classification result. This process is shown in Figure 5.3

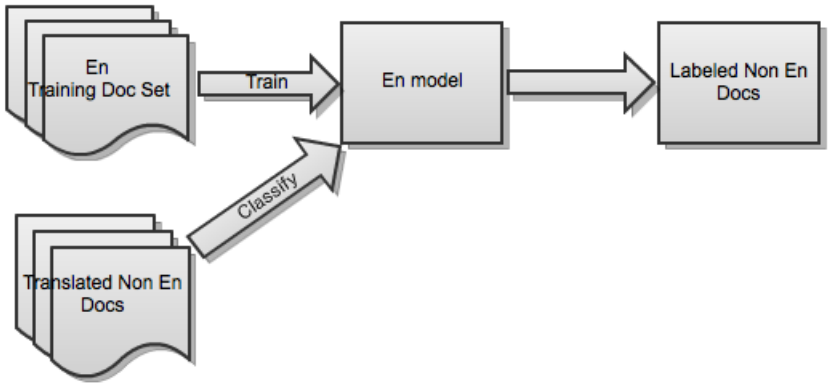

Figure 5.3: Traing the English Documents and Label the Non English training set

After training and classifying with the document translation method, we have a set of labeled documents in other languages. Those labeled documents will go through the feature selection algorithm. The extracted feature is combined with the feature from English documents, and there is a merge factor which is defined as how many percentage of feature will be merged into the English features. Duplicated feature are counted only once. The mixed features set is translated into each of the different target languages.

For the set of test documents in Non English documents, based on the translated feature list, a feature table is generated for every document. If the feature exists in the document, then a 1 is recorded in the feature table, otherwise, 0 will be recorded.

With mixed feature sets, the feature table for the English document should be refreshed as well, and the refreshed feature table is used for the SVM^{light} to train a model.

After the feature table has been calculated for all the documents in Non

English test documents set, the feature table file is going to be fed to SVM^{light} as a classification input file.

SVM^{light} trains a new model base on the feature table file generated from mixed feature set, then the model is used to classify the feature table file from the Non English test document set. The process of cross lingual classification is shown in Figure 5.4. The classification result is shown in Table 5.2, Table 5.3 and Table 5.4.

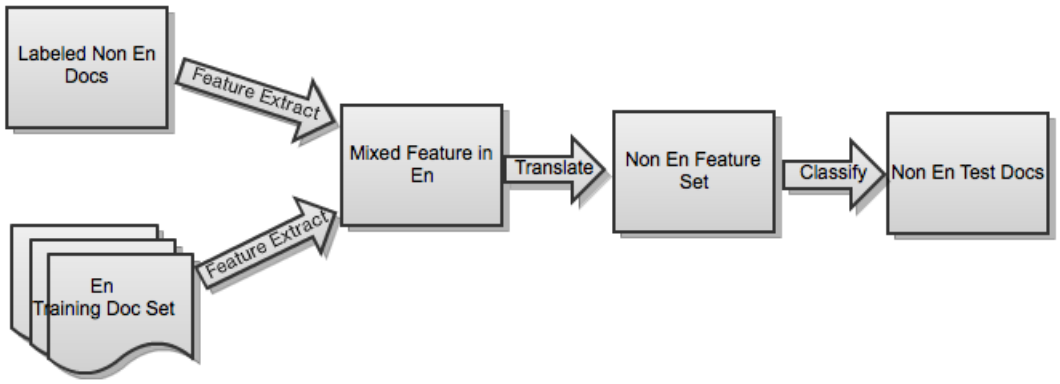

Figure 5.4: Cross Lingual Classification with mixed feature set from document translation

5.3 Naïve Bayes

5.3.1 Introduce Naïve Bayes Classifier

The Naïve Bayes classifier is based on the classic mathematic theory, and has a concrete mathematical foundation, as well as a stable classification performance.The number of parameters used for the Naïve Bayes to classify is small, and it is insensitive to data lost, the algorithm is simple. It is very useful as a base line test classifier. The Naïve Bayes model in theory should have a smaller error rate, but due to the strong assumption of feature independency, the performance is not always as good as the theory.

Naïve Bayes is base on Bayes Theorem which can be defined with following formular:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$P(Y|X)$ denote when event X happens, the probability of event Y happens.

The reason why Bayes Theorem is so popular and useful is because we always have this kind of problem: we know $P(X|Y)$, but $P(Y|X)$ is hard to get, but we need $P(Y|X)$, Bayes Theorem lead us a way from $P(X|Y)$ to $P(Y|X)$.

Table 5.2: Classification Results for Spanish with SVM

	En	Es_doc	Es_mdl	Mix
Business	96.49%	84.17%	79.99%	87.05%
Entertainment	94.12%	83.02%	71.65%	79.42%
Health	95.97%	89.32%	52.68%	89.03%
Sports	97.38%	89.02%	90.32%	86.16%

Table 5.3: Classification Results for French with SVM

	En	Fr_doc	Fr_mdl	Mix
Business	96.49%	89.02%	79.67%	89.35%
Entertainment	94.12%	87.37%	78.77%	87.41%
Health	95.97%	82.58%	83.05%	87.65%
Sports	97.38%	92.57%	91.36%	91.73%

Table 5.4: Classification Results for Chinese with SVM

	En	Ch_doc	Ch_mdl	Mix
Business	96.49%	92.17%	84.83%	86.93%
Entertainment	94.12%	91.17%	92.33%	92.37%
Health	95.97%	82.77%	76.77%	90.17%
Sports	97.38%	94.47%	82.73%	89.67%

When using Bayes Theorem in text classification, we assume $x = \{a_1, a_2, ..., a_n\}$ is one of the item that waiting for classify, each a is one feature of x, category denote as $C = \{y_1, y_2, ..., y_n\}$, then calculate $P(y_1|x), P(y_2|x), ..., P(y_n|x)$, if $P(y_k|x) = max\{P(y_1|x), P(y_2|x), ..., P(y_n|x)\}$, then $x \in y_k$.

If all the feature is independent, base on Bayes Theorem:

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

Because the denominator for all class is a const number, so we only need to maximum numerator:

$$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i)...P(a_n|y_i)P(y_i) = P(y_i)\Pi_{j=1}^{m}P(a_j|y_i)$$

5.3.2 Naïve Bayes Classifier in Weka

Weka [HFH+09] is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Weka is open sourced, and it comes with an implementation of the Naïve Bayes classifier.

There are three stages that are part of using the Naïve Bayes classifier in Weka.

In the first stage known as data preprocessing, documents go through feature selection algorithms, and a subset of the feature with a higher value are selected. I only recorded if the feature is present or not, instead of giving a weighted value from the feature selection algorithm. Because Naïve Bayes Bayes is a multi-class classifier, the documents can be used in one chunk.

In the second stage, attributes are assigned. We then call Weka API to train Naïve Bayes Bayes classifier.

In the Third stage, the test document goes through the feature selection algorithm, and call Weka API to classify the test documents.

The classification result for each language are shown in Table 5.5, 5.6, 5.7.

5.4 Result Analysis

Table 5.2, table 5.3 and table 5.4 show the classification accuracy with SVM classifiers for each different methods. First column is the class names, second column is the classification result with pure English documents, third column is the classification accuracy of the document translation method, fourth column is the classification accuracy of the model translation method and the last column is the

Table 5.5: Classification Results for Spanish with NB

	En	Es_doc	Es_mdl	Mix
Business	94.09%	42.67%	19.43%	52.18%
Entertainment	70.56%	99.60%	63.64%	84.91%
Health	90.08%	69.25%	36.93%	72.10%
Sports	92.08%	74.20%	10.13%	70.60%

Table 5.6: Classification Results for French with NB

	En	Fr_doc	Fr_mdl	Mix
Business	94.09%	66.67%	55.53%	68.25%
Entertainment	70.56%	93.27%	66.07%	81.01%
Health	90.08%	55.70%	21.63%	57.20%
Sports	92.08%	68.07%	16.20%	50.99%

Table 5.7: Classification Results for Chinese with NB

	En	Ch_doc	Ch_mdl	Mix
Business	94,09%	61.33%	16.07%	45.09%
Entertainment	70.56%	98.80%	58.27%	86.81%
Health	90.08%	49.27%	37.40%	33.09%
Sports	92.08%	77.33%	6.40%	69.00%

classification result of the proposed approach. From the tables, we learn that pure English documents has the best accuracy, document translation method has second high accuracy, the proposed approach has better accuracy then the model translation method as we expected. For business documents in Spanish, business, entertainment and health documents in French, entertainment and health documents in Chinese, the proposed approach even better then the document translation method. The result shows a strong evidence that the features from document translation method is boosting the classification accuracy of the model translation method.

Table 5.5, table 5.6 and table 5.7 show the classification accuracy with Naïve Bayes classifiers for each different methods. The result table has same structure with the result table for SVM. With same method and data, Naïve Bayes classifier has lower accuracy then SVM. Also it is very sensitive to the model translation method. Our proposed approach can increase the classification accuracy for the model translation method, further more, for business and health documents in Spanish, entertainment and health documents in French, the proposed approach even has better accuracy then the document translation method.

Overall, we can see that the proposed approach has better classification results than the model translation method in most cases. Indeed, sometimes it is even better than the document translation method. As we mentioned before, the document translation is a slow and expensive operation, my proposed method has better performance in both time and lower cost benefit as well.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this paper, I proposed a new method which combines the document translation and model translation features table together to improve the Cross Lingual Text Classification result. I also built the data set of cross lingual from scratch. I discussed the different feature selection algorithms and impacts to the classification result. I pointed out a strategic method to classify the CJK documents without doing tokenization. I identified the performance bottleneck and utilized multi-threading to improve it.

The result of classification shows that, compared to using only the model translation method, the proposed approach which is to use the model translation method combined features with document the translation method can achieve better performance. When we compare the proposed approach to the document translation method, the proposed approach has lower accuracy for some cases, however, the proposed method does not require the user to translate all documents, thus saving additional time and cost. Sometimes the proposed approach has better accuracy then the document translation method. The proposed approach has better accuracy then the model translation method for most cases.

6.2 Future Work

When extracting features from translated documents which has been classified by the English model, there are always some documents that are mis-classified. Even though the number of documents are small and due to the nature of feature selection algorithm run later on to the features in those documents, the feature in those false positive cases have a chance to get into the mixed feature set, then lead to mis-classification of documents in other languages. The future work to target would be to detect and predict the confidence of the classification result of the document translation method. Only then can we select and mix those features from the documents with a high confidence.

BIBLIOGRAPHY

[AM07] B.M. Amine and M. Mimoun, *Wordnet based cross-language text categorization*, Computer Systems and Applications, 2007. AICCSA '07. IEEE/ACS International Conference on, 2007, pp. 848 –855.

[bag] *Bag of word theory, http://en.wikipedia.org/wiki/bag_of_words_model*.

[DDH+07] Anirban Dasgupta, Petros Drineas, Boulos Harb, Vanja Josifovski, and Michael W. Mahoney, *Feature selection methods for text classification*, Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (New York, NY, USA), KDD '07, ACM, 2007, pp. 230–239.

[Fel98] Christiane Fellbaum, *Wordnet: An electronic lexical database.*, MIT Press, 1998.

[Fun06] Apache Fundation, *http://lucene.apache.org/solr/*, 2006.

[HFH+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Peutemann, and Ian H. Witten, *The weka data mining software: An update*, SIGKDD Explorations, 2009.

[LXD+08] Xiao Ling, Gui-Rong Xue, Wenyuan Dai, Yun Jiang, Qiang Yang, and Yong Yu, *Can chinese web pages be classified with english data source?*, Proceeding of the 17th international conference on World Wide Web (New York, NY, USA), WWW '08, ACM, 2008, pp. 969–978.

[LXZH09] Shoushan Li, Rui Xia, Chengqing Zong, and Chu-Ren Huang, *A framework of feature selection methods for text categorization*, Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2 (Stroudsburg, PA, USA), ACL '09, Association for Computational Linguistics, 2009, pp. 692–700.

[Mil95] George A. Miller, *Wordnet: a lexical database for english*, Communications of the ACM (New York, NY, USA), vol. 38, ACM, November 1995, pp. 39–41.

[NSHC11] Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen, *Cross lingual text classification by mining multilingual topics from wikipedia*, Proceedings of the fourth ACM international conference on Web search and data mining (New York, NY, USA), WSDM '11, ACM, 2011, pp. 375–384.

[PS10] Peter Prettenhofer and Benno Stein, *Cross-language text classification using structural correspondence learning*, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (Stroudsburg, PA, USA), ACL '10, Association for Computational Linguistics, 2010, pp. 1118–1127.

[SMT10] Lei Shi, Rada Mihalcea, and Mingjun Tian, *Cross language text classification by model translation and semi-supervised learning*, Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (Stroudsburg, PA, USA), EMNLP '10, Association for Computational Linguistics, 2010, pp. 1057–1067.

[TJBe99] B. Schölkopf T. Joachims, C. Burges, and A. Smola (ed.), *Making large-scale svm learning practical. advances in kernel methods - support vector learning*, MIT-Press, 1999.

[Vap95] Vladimir N. Vapnik, *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[Wan09] Xiaojun Wan, *Co-training for cross-lingual sentiment classification*, Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1 (Stroudsburg, PA, USA), ACL '09, Association for Computational Linguistics, 2009, pp. 235–243.