Text Classification Based on Background Knowledge

Chao Li

A Thesis submitted to the University of Tokushima in partial fulfillment of the requirements for the degree of Doctor of Philosophy

2017

Tokushima University
Graduate School of Advanced Technology and Science
Information Science and Intelligent Systems

Contents

List of Figures

List of Tables

Acknowledgment

First, I would like to thank my advisor Professor Ren Fuji. With his kind help and support I was able to continue my study on the basic theories of machine learning and information retrieval. He created pleasant research environment for me and provided me with a Linux server of high standard configuration. He supported me in doing large experiments which took long time and permitted so many failures in these experiments.

I would also thank the members of A1 group for their support and encouragement. Our discussions on works and papers gave me much inspiration in my study. Thanks to Xin Kang, Ning Liu, Duo Feng, MengJia He and Zhao Han.

I would thank the Department of Information Science and Intelligent Systems and all the professors in this department, who make up a conscientious and careful research community. They are very kind and patient to foreign students and helped me a lot when taking my doctoral courses. Specifically, I would thank Professor Kenji Kita, and Professor Masami Shishibori, who had contributed so much time and efforts in reviewing this thesis. Their careful analysis and recommendations helped to improve this thesis.

Finally, thanks to all the Japanese language teachers in Tokushima university.

Abstract

Nowadays, information technology is developing fast and used in more and more fields. Therefore, numerous documents are saved in the computers which could be read by the computers. Moreover the number of the documents increases extremely everyday. For the important application, it becomes a research subject how to automatically classify, organize and manage such numerous amount of literature and data, in the case most of them are documents.

Due to the growing availability of digital textual documents, automatic text classification (ATC) has been actively studied to organize a vast number of unstructured documents into a set of categories, based on the textual contents of the document.

Text representation is the fundamental step in text classification task, in which a text is represented by a set of features. Features play the important roles in training classification model and prediction. Many previous studies focused on enriching text representation to address text classification task. However, the traditional classification methods with VSM (Vector Space Model) only studied intensively on the words and their relationship in some specific corpus/dataset.

According to the previous researches, the key problem of text classification is the lack of information, especially for the imbalanced dataset.

In this thesis, we propose the idea of the background knowledge, which could complement information for documents and train models to classify texts.

This study is based on Baidu Baike and character co-occurrence. Baidu Baike is an online Chinese encyclopedia similar to Wikipedia, which is widely used by Chinese speakers to learn basic concept and general knowledge. Two external corpora are employed for extracting the features of character co-occurrence, People's Daily news and Sougou news corpus.

To predict the categories for new texts, SVM (Support Vector Machine), a machine learning algorithm, is used to train the classification models. The performance of proposed method is evaluated with Fudan University text classification corpus and Sougou classification corpus in reduced version.

The results show that the background knowledge could complement the information for the documents in text classification task. With the imbalanced corpus, the improvement is obvious by adding the background knowledge.

Chapter 1

Introduction

Because computers and Internet are widely used, the extremely huge number of information is produced everyday. Nowadays, the information already is full of our life. Most of the information is stored as texts. A large number of unstructured texts is posted and sorted in web pages, digital libraries and community. Therefore, the automatic method is necessary to help people manage and filter these information instead of manual work.

Predicting the class labels for the online texts has been required by a variety of applications. For example, in spam filtering, classification methods are used to determine the junk information automatically. In news organization, because most news is provide on Internet and the amount is huge, it is impractical to finish this task manually. In emotion classification, a text is classified by the emotion based on their meaning. In recommendation systems, the class of the text would be an important tag, which determines if its content might draw more attention to this customer.

1.1 Text Classification and Background Knowledge

Depending on the classification task, there are different kinds of class sets. For example, in spam filtering, there are 2 classes in this task, which is a binary classification problem. It only determines if the text is spam mail or not. In news organization, on CNN.com, the news channels include *money*, *entertainment*, *tech*, *sport*, *travel*, and so on. In emotion classification, there are *Joy*, *Love*, *Expectation*, *Surprise*, *Anxiety*, *Sorrow*, *Anger*, and *Hate* defined in Ren-CECps [29].

The motivation for exploiting background knowledge in text classification is attributed to two reasons. First, more information from texts can make more reasonable classification. Second, people have basic concepts and general knowledge in their mind, however, the common corpora/datasets are some kinds of special case which would lack some basic concepts and general knowledge. These basic concepts and general knowledge are the background knowledge in our life.

1.1.1 Text Classification

The goal of text classification is predicting the correct class label for a given text. Text classification task is defined as a set of training texts D=$\{X_1, ..., X_n\}$, each text is labeled with a category value drawn from a set of k different discrete values which are indexed by {1, ..., k} [1]. All the texts are split into 2 subsets, training texts and test texts. The training texts are used to train classification model by using machine learning algorithms. The test texts are used to evaluate the performance of the model. For a test text whose category is unknown, the model is used to predict its category. Each category is assigned with a label. These labels are numeric values that represent the categories. Practically, text classification task is computing the text's label. For example, in the following sentence:

He played basketball yesterday.

The word *basketball* indicates that this text is related to *sport.* If *sport* is labeled by 1, the computer should give us 1 as the result.

For another example:

He played basketball at P.E. class yesterday.

The word *class* indicates that this text is also related to *education.* In this case, it is difficult to determine the correct label.

In general, there are several categories in a corpus. The text classifier has to determine the category for each text.

The good classifier should try its best to predict the labels for texts with high precision, and also include the possible text categories. To achieve more accurate prediction, in this thesis, we use the background knowledge to enrich the text representation.

1.1.2 Background Knowledge

One corpus/dataset is just a part of universal set. Therefore, sometimes the information in the training texts is not enough to predict the class labels for the test texts, which would reduce the accuracy. In the extreme case, there is not any similar information between the training and test texts, such as the imbalanced corpus, in which the size of some categories is much smaller than others. When dealing with the classification problem, people first search the concepts and knowledge stored in their mind, and compute similarities to make decision. Similarly, this study assumes that is the computer also can work better for classifying texts with the background knowledge.

The inspiration of background knowledge is from some previous researches. Ren (2010) proposed the idea of employing "cloud computing" in NLP [32]. According to this idea, the external unlabeled texts, as the background knowledge, add the new information to each text for classification task.

Online encyclopedia

An encyclopedia is a type of reference work or compendium holding a comprehensive summary of information from either all branches of knowledge or a particular branch of knowledge.[1] An online encyclopedia is an encyclopedia accessible through the internet.[2]

For example, Wikipedia is a well-known online encyclopedia. It has around 5 millions articles in English. And a lot of users keep contributing their work to add the articles and materials.

For another example, Baidu Baike, which is similar to Wikipedia, is an online ency-

[1] https://en.wikipedia.org/wiki/Encyclopedia

[2] https://en.wikipedia.org/wiki/Online_encyclopedia

clopedia in Chinese language. It has around 14 millions articles in Chinese. It is widely used by Chinese speakers to learn basic concepts and general knowledge.

In this thesis, we complement information to texts based on Baidu Baike for text classification in Chinese. With the information extracted from Baidu Baike, our method can overcome the lack of similar information between training and test texts. For example, there two sentences:

Training text: *He likes playing basketball.*

Test text: *I love this game.*

There are not any same words in these two sentences. We can determine the 1st sentence is related to sports easily. However, if we don't have any background knowledge, we can not know the 2nd sentence is also related to basketball. Because it is a famous slogan for NBA (National Basketball Association). This shows the effect of background knowledge in text classification.

Character Co-occurrence

In English, characters are the 26 letters, A-Z. Each letter has its pronunciation but does not have meaning. These characters spell words and tell people how to pronounce the words. Words have meanings.

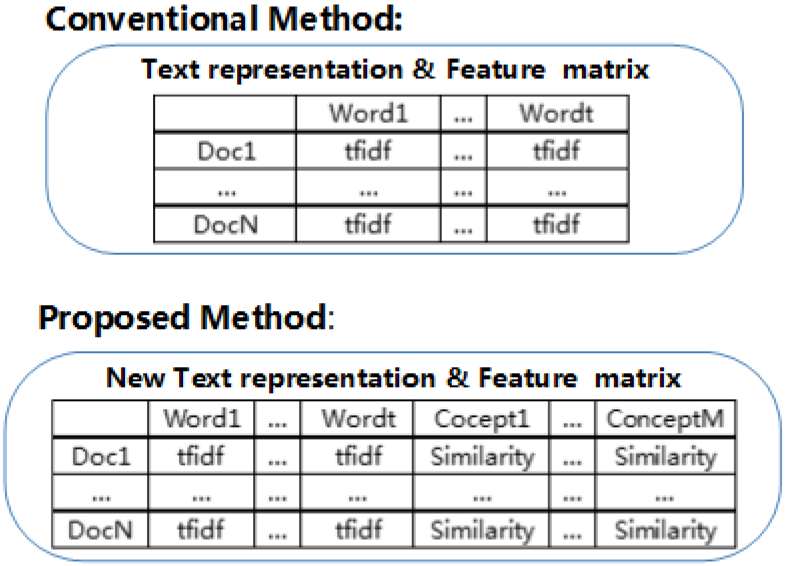
In Chinese, characters are not letters, they are named Hanzi. Each Hanzi has its pronunciation. And, each Hanzi has its meaning. It is similar to English, Hanzi spells words and tells people how to pronounce the words. However, the difference is that single Hanzi is also used as a word to express meaning sometimes. For examples, 马 *(mǎ)* means *horse* in English. 跑 *(pǎo)* means *run* in English. 快 *(kuaì)* means *hurry* in English. In these examples, these words include a noun, a verb and an adjective. All of them can express by the single Hanzi. This is the reason that we focus on Chinese characters in this thesis.

Based on English, some previous researches presented the word co-occurrence. The word co-occurrence means some words often appearing in a sentence or text together, which can imply the meaning/semantics of the sentence or text. With our experience, some word co-occurrence means the specific meaning in some regularity. Therefore, word co-occurrence can be regarded as the background knowledge. Because the character/Hanzi can express meaning by itself in Chinese. Thus, we investigate the Chinese characters co-occurrence in this thesis.

1.2 Overview of Methods and Contributions

1.2.1 Acquiring Background Knowledge

In this study, we use the articles from Baidu Baike and the character co-occurrence from unlabeled corpora as the background knowledge. In Baidu Baike, each article describes a concept. The articles from Baidu Baike are regarded as the background knowledge. By using Baidu Baike, we propose the text representation with the similarities between the texts and articles from Baidu Baike. Figure 1.1 shows the difference between the proposed method and the traditional method.



Conventional Method:

Text representation & Feature matrix

	Word1	...	Wordt
Doc1	tfidf	...	tfidf
...	...	...	...
DocN	tfidf	...	tfidf

Proposed Method:

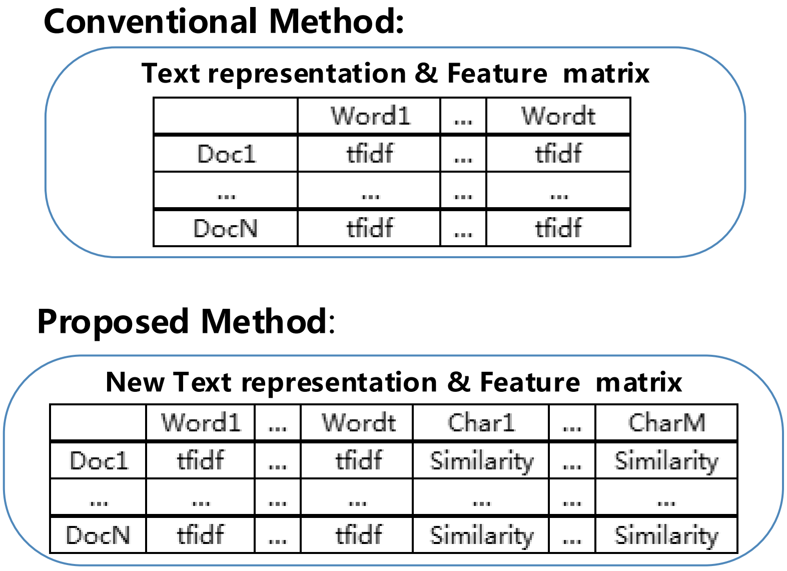
New Text representation & Feature matrix

	Word1	...	Wordt	Cocept1	...	ConceptM
Doc1	tfidf	...	tfidf	Similarity	...	Similarity
...	...	...	...	...	...	...
DocN	tfidf	...	tfidf	Similarity	...	Similarity

Figure 1.1: Difference between the proposed method and the conventional method. $tfidf$ is the TF-IDF weighting, 'Similarity' is the similarity between the text and the article from Baidu Baike.

For each category, some keywords are selected to represent their categories. To obtain the keywords for each category, a TF-IDF liked method is proposed to weight and rank the words, called CTF-ICF, which works on the category level, while TF-IDF works on text level. To obtain the articles of the keywords, a program is implemented to search the concepts and download the articles from Baidu Baike. With the articles from Baidu Baike, the performance is evaluated on different proportion of training and test texts. The results show that the performance is improved by using the background knowledge. Specially, it is robust for imbalanced corpus, in the case of small size of training texts.

To study character co-occurrence, we propose the text representation with the similarities of character co-occurrence between texts and unlabeled corpora. The unlabeled corpora are used to count the frequency of character co-occurrence. The frequency of character co-occurrence is regarded as the background knowledge. Figure 1.2 shows the difference between the proposed method and the traditional method.



Conventional Method:

Text representation & Feature matrix

	Word1	...	Wordt
Doc1	tfidf	...	tfidf
...	...	...	...
DocN	tfidf	...	tfidf

Proposed Method:

New Text representation & Feature matrix

	Word1	...	Wordt	Char1	...	CharM
Doc1	tfidf	...	tfidf	Similarity	...	Similarity
...	...	...	...	...	...	...
DocN	tfidf	...	tfidf	Similarity	...	Similarity

Figure 1.2: Difference between the proposed method and the conventional method. $tfidf$ is the TF-IDF weighting, 'Similarity' is the similarity between the text and the unlabeled corpus.

To study the impact of different background knowledge, two external corpora are used as the background knowledge, People’ s daily news and SougouCA news. For each category, some key characters are selected to represent their categories, in order to reduce the computation. The results show that the performance outperforms the traditional method with the background knowledge. Specially, it obtains better performance for the imbalanced corpus.

1.2.2 Training Classification Model

To train classification model, the support vector machine (SVM) classifier is used in the experiments, which is a geometric model. SVM is a supervised learning models for classification and regression task. The training data and test data has to be labeled before using SVM. And then, each example is represented by a group of features. These features usually are numeric values. After extracting the features, the examples are usually represented as linear vectors in very high dimensions. The dimensions corresponds to the features. Finally, with different kernel function, SVM can deal with linear and non-linear classification task.

For example, the bag-of-words (BoW) model use the frequency of words to represent sentences or documents. Each sentence or document is converted to the vector of word frequency. Each index represents a word. The length of the vector is the same as the size of the vocabulary. The vocabulary includes the words which appear in all the sentences or documents. And then, these feature vectors are used to train the SVM classifier for predicting the new examples.

In this thesis, TF-IDF algorithm is used to represent texts, instead of BoW model. By using TF-IDF algorithm, the length of feature vectors also equals the size of the vocabulary. Each index represents a word. But each value in the feature vectors is computed with TF-IDF algorithm rather than the word frequency. Besides the TF-IDF algorithm, the background knowledge is also used to complement the information for the text representation, as showed in Figure 1.1 and 1.2. The similarities between the texts and the background knowledge are added to the feature vectors for training the SVM classifiers. Some other details will be introduced in the following section.

1.2.3 Other Machine Learning Methods

Most traditional machine learning algorithms, including Decision Trees, Support Vector Machines (SVM), Naive Bayes (NB) and Neural Network [1], are designed to learn and predict a single label or a sequence of labels for texts.

In this thesis, we compare the results of traditional method without background knowledge and the proposed method with background knowledge. To make sure the results from traditional method and the proposed method are comparable, we use the same classifier for these models with the different groups of features. The results are evaluated by the precision and recall score. And we also make further discussion on these results.

1.3 Thesis Organization

This thesis covers the background, related work, methods, results, and discussions about predicting text labels with the background knowledge, which is organized in the rest chapters as follows.

Chapter 2: Background

In this chapter, we begin by reviewing the text classification task. We illustrate the basic framework of text classification and present the overview of the workflow. Then we focus on the different kinds of features for representing features, and review the methods for extracting the features. We also introduce some machine learning methods used in text classification.

Chapter 3: Related Work

In this chapter, we review the study of text classification about enriching text representation with additional source. First, we introduce some related work about extracting features from texts. Second, we discuss some methods about dealing with the imbalance dataset of text classification. Third, some Wikipeida based method are surveyed. We regards the Wikipeida as the knowledge base which can provide the background knowledge. Inspiring by these idea, we propose our methods. Finally, we present a term weighting method. From this idea, we propose CTF-ICF method to select keywords for each category.

Chapter 4: Extracting Background Knowledge for Text Classification

In this chapter, we discuss the methods that we have proposed for enriching the text representation for text classification. We first introduce the method based on Baidu Baike. The proposed method is illustrated with the workflow. And then, we introduce the details for each step. The articles from Baidu Baike are used as the background knowledge in this method. To select keywords for each category, we proposed the CTF-ICF method. To download the articles, a program is developed to extract the search results from Baiku Baike. The text representation is enriched with the similarities between the texts and the articles. Second, We introduce the method based on character co-occurrence. The proposed method is also illustrated with its workflow. And then, The details for each step are introduce. The texts from the additional corpus are used as the background knowledge by counting the character co-occurrence. The key characters are selected for representing their categories and reducing the computation. The text representation is enriched with the similarities of the character co-occurrence between the texts and the additional corpus. Finally, the evaluation methods and tools used in the experiments are introduced.

Chapter 5: Evaluation

In this chapter, we present the experimental results and discuss the findings. In the first part, the experimental results of the proposed method based on Baidu Baike are presented. In the second part, the experimental results of the proposed method based on character co-occurrence are presented. In both parts, we first illustrate the classification corpus and the setting in the experiments. To evaluate the method based on Baidu Baike, we compare a baseline method, the traditional method, and the proposed method with different numbers of keywords. To evaluate the method based on the additional corpus, we compare the experiments with different classification corpora, additional corpora, numbers of keys characters, and distance between words. Finally, we discuss some findings according to the results.

Chapter 6: Contribution and Recommendation

In this chapter, we conclude the contributions of this thesis and discuss the directions for future study. We find that the text classification is still a challenging task. To improve the classification models, we can discovering more context sensitive features, feature selection methods, as well as more appropriate machine learning methods.

Chapter 2

Background

In this chapter, we first review the methodologies for text classification. Concretely, we illustrate the framework of text classification and overview. The framework includes text representation, training model and prediction. For text representation, we review some kinds of text features. For training model, some machine learning methods were introduced for text classification. Lastly, we review some other methods for preprocessing.

2.1 Methodologies of Text Classification

The goal of text classification is predicting the correct class label for a given text. This is a supervised classification task in which each text must be labeled before training model and prediction. This task is defined as a set of training text D=$\{X_1, ..., X_n\}$, each text is labeled with a category value drawn from a set of k different discrete values which are indexed by {1, ..., k} [1].

2.1.1 Framework of Text Classification

The main steps of text classification include text representation, training classification model and predicting class label. For example, predicting the topic of a news article from a fixed list of topics such as ”game”, ”technology” and ”travel”. The framework of text classification used by supervised classification is shown in Figure 2.1.

During training, the input text is converted into a group of features by using a feature extractor. The details of feature extractors are discussed in the section 2.2. The machine

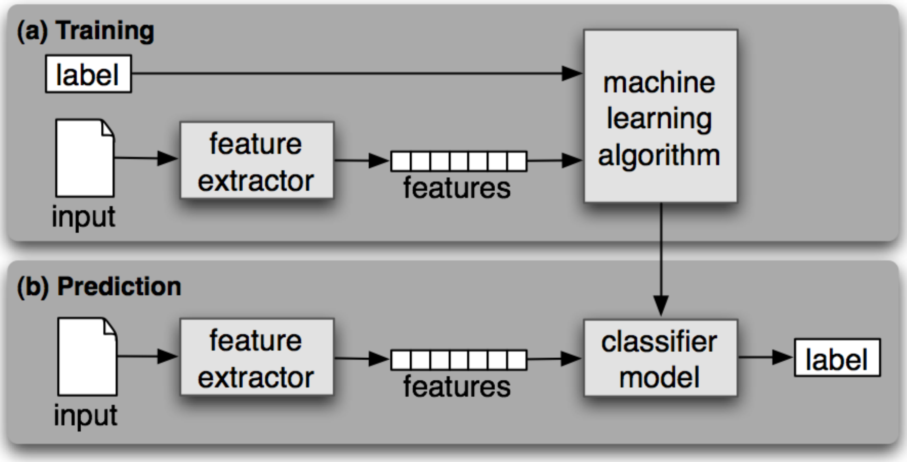


Figure 2.1: The framework of text classification [3]

learning algorithm use the pairs of text features and labels to training the classification model. During prediction, the new text is also converted into a group of features by the same feature extractor. The model use these features to predict the class label for this text.

2.1.2 Overview of Text Classification

The entire steps and elements include collecting corpus, selecting instances, resampling, extracting features, weighting, selecting features, training model, predicting, and evaluating performance.

The corpus is most important for text classification. With the corpus having exact labels, we can training the exact classifiers. Whereas we can not get the good classifiers with inexact labels, despite using the best methods. Commonly, researchers choose opened standard dateset to evaluate their methods.

In English, these datasets include 20 Newsgroups, Reuters-21578, WebKB and RCV1-v2/LYRL2004 which are widely used benchmark collections in text classification task.[1,2] The 20 Newsgroups has 20 categories and 18,821 documents. The Reuters-21578 has 2 versions, R8 and R52. The R8 has 8 categories and 7,674 documents. The R52 has 52 categories and 9,100 documents. The WebKB has 4 categories and 4,199 documents. The RCV1-v2/LYRL2004 has 103 categories and 804,414 documents from four parent topics.

[1] http://www.cs.umb.edu/ smimarog/textmining/datasets/
[2] http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

In Chinese, there are FUDAN University text classification corpus[3,4] and Sougou text classification corpora (SogouCA[5] and SogouCS[6]). FUDAN University text classification corpus has 20 categories and 19,637 documents. The SogouCA is 1.02GB in tar.gz file. The SogouCS is 65GB in tar.gz file. In these two corpora, the number of the categories depends on the websites where the news are from.

Selecting instances chooses the texts which are more discriminable than others for training models. In other words, selecting instances filters out the noisy texts which reduces the accuracy for classifiers. Resampling is a little similar to selecting instances. The difference is that resampling remove and create some texts for each category in the corpus. Extracting features is also an important step, besides the corpus. Good features can get good classifiers. Therefore, many work focused on how to extracting better features for training model. Weighting often goes along with features. Depending on the features, many weighting methods have been proposed. After weighting features, sometimes, selecting features filters out the noisy features which have less effort for classifying texts. This is similar to selecting instances, but selecting features works on the feature level. To training classifiers, commonly, the machine learning methods are used. With the trained classifier, we can predict the class label for a new text. Finally, The performance of the classifier should be evaluated and presented to other researchers and engineers. The evaluation results tell them how the proposed methods can work and if there are any help for their work.

It is regard to these steps, many researches had been conducted on text classification in recent years. The previous work include instance selection [45][49][46], resampling methods [7][16][51][28][36], feature selection [8][22][38][12][13][23], weighting method [21][39][34][17], kernel function [19], ensemble of features and algorithms [52][26].

Instance Selection

To filter out noisy texts from the training texts, Wang et al. (2013) proposed the Border-Instance-based Iteratively Adjusted Centroid Classifier (IACC_BI), which uses the border

[3] http://www.datatang.com/data/44139

[4] http://www.datatang.com/data/43543

[5] http://www.sogou.com/labs/resource/ca.php

[6] http://www.sogou.com/labs/resource/cs.php

instances to construct centroid vectors for the centroid-based classifier [49]. In 2013, Tsai and Chang (2013) proposed the support vector oriented instance selection (SVOIS) to select the training texts, based on the support vectors [45]. In 2014, Tsai et al. (2014) proposed the biological-based genetic algorithm (BGA) to reduce the training texts, which simulates the evolutionary process [46].

Resample

In text classification, the number of texts in the categories is different. In this case, the rare categories have less texts than others. To balance the text distribution, Chen et al. (2011) generated new texts for rare categories to resample the examples based on probabilistic topic models [7]. Wang et al. (2013) used the boundary region cutting (BRC) algorithm to remove samples for imbalanced text sets [51]. Iglesias et al. (2013) proposed the content-based over-sampling HMM (COS-HMM) to generate new texts according to current text [16] for the class imbalance problem. Qian et al. (2014) proposed a resampling ensemble algorithm in which the small categories are oversampled and large categories are undersampled [28]. Sáez et al. (2015) proposed the SMOTE-IPF method to filter out the noisy and borderline examples. the SMOTE-IPF is an extension of SMOTE (Synthetic Minority Over-sampling Technique). The IPF represents Iterative-Partitioning Filter.

Feature Selection

To filter out the noisy features, Covões and Hruschka (2011) proposed a filter-based algorithm by splitting the set of features into clusters [8]. And the features are selected based on feature-class correlations. Maldonado et al. (2011) proposed the kernel-penalized SVM (KP-SVM) method, which selects relevant features during training SVM classifier [22]. Shang et al. (2013) proposed the global information gain (GIG) and the maximizing global information gain (MGIG) methods based on information gain (IG) for selecting features. Guan et al. (2013) focused on reducing the computational overhead of Singular Value Decomposition (SVD) with Latent Semantic Indexing (LSI) [12]. They used Spectrum Analysis (ISA) to reduce dimension fast. Hrala and Král (2013) evaluated five feature selection methods with Czech corpus and stated that Maximum Entropy and SVM outperform other methods [13]. Maldonado et al. (2014) focused on imbalanced data sets.

They proposed a family of methods based on a backward elimination approach for ranking and selecting features [23].

Weighting Method

In text classification, the weight correspond to the contribution for representing text. Features with high weight mean they are important. Features with low weight mean they are less important. To weight features, Christina Lioma and Roi Blanco (2009) proposed a POS (Part of Speech) n-gram method to weight features [21]. Shi et al. (2011) proposed an improved TF-IDF weighting method by adding concentration and dispersion information [39]. Ren and Sohrab (2013) proposed the class-indexing-based term-weighting methods, called TF.IDF.ICF and TF.IDF.ICS$_\delta$F, which are the extension of TF-IDF method [34]. The ICF is the inverse class frequency. The ICS$_\delta$F is the inverse class space density frequency. Jiang et al. (2016) proposed the deep feature weighting (DFW) method to compute feature weighted frequencies from training text [17]. With the naive Bayes model, this method is used to estimate the conditional probabilities.

Kernel Function

Kim et al. (2014) proposed a kernel method, called language independent semantic (LIS) kernel, to compute the similarities between short-text documents based on semantic annotations [19].

Ensemble Method

Xia et al. (2011) used the part-of-speech based features and the word-relation based features to train classifiers with naïve Bayes, maximum entropy and support vector machines [52]. Then three ensemble strategies are used to predict the class labels, namely the fixed combination, weighted combination and meta-classifier combination. Onan et al. (2016) investigated the ensemble of five statistical keyword extraction methods and four learning algorithms by using five ensemble methods [26]. The keyword extraction methods include most frequent measure based keyword extraction, term frequency-inverse sentence frequency based keyword extraction, co-occurrence statistical information based keyword extraction, eccentricity-based keyword extraction and TextRank algorithm. The learning algorithms include Naı̈ve Bayes, support vector machines, logistic regression and Random Forest. The ensemble methods include AdaBoost, Bagging, Dagging, Random Subspace and Majority Voting.

Features and Algorithms

To extract features from text, many kinds of features have been presented, such as words, text structure [2], part-of-speech [10], n-gram [40], syntactic and semantic feature [5][54], phrase pattern [53], feature transformation [47], and Neural Networks [20]. In the next section, we will introduce these features.

Besides, many machine learning methods have been used for text classification, including Naı̈ve Bayes, support vector machines, logistic regression, centroid, K-nearest neighbor (KNN), neural network, k-means, clustering, active learning and classifier combination [18][27][9][14][15][26]. These methods will be introduced later.

2.2 Extracting Features for Text

In the section, we introduce several text features and the extraction methods. Text features are the key aspect for text classification task. Therefore, extracting reasonable features is an important. Machine learning methods uses text features to training classifiers. The performance of classifiers depends on the discrimination of features. With good features, we can train good classifiers.

2.2.1 Words

In most natural language processing problems, words are the common features for texts. For example, the bag-of-word (BoW) model as mentioned above, each sentence or document is converted to the vector of word frequency and each index represents a word.

Instead of BoW model, TF-IDF algorithm is used to represent text commonly. Words are used as the features weighted by TF-IDF algorithm [37][30], referring to the Equations 2.1-2.3. Each text is represented by a vector of numeric value. Each index corresponds to a word. The length of the feature vector equals the size of vocabulary. The vocabulary includes the words from all texts. All the texts could form the feature matrix. Then, with the feature matrix, the classifiers are trained to predict the class labels.

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{2.1}$$

where n_{ij} is the frequency of word i in text j, $\sum_k n_{ij}$ is the number of all words in text j.

$$IDF_i = \log \frac{|D|}{|D(word_i)|} \tag{2.2}$$

where $|D|$ is the number of texts in the corpus, $|D(word_i)|$ is the number of texts which contain word i.

The discriminability of word i to document j could be weighted as following:

$$w_{ij} = TF_{ij} \times IDF_i \tag{2.3}$$

2.2.2 Structure

Alías. (2008) represented texts as a directional weighted word-based graph based on associative relational network [2]. The associative relational network (ARN) is a graph model. Figure 2.2 shows the Word-based associative relational network. The nodes represent words. The edges means the co-occurrence relation between words. Each node and edge has its weight. Each node is weighted by TF-IDF and inverse term frequency (ITF) methods, referring to the Equation 2.4.

$$ITF_i^k = log(\frac{|\tau^k|}{TF_i^k}) \tag{2.4}$$

Where $|\tau^k|$ represents the number of words/terms, and TF_i^k is the frequency of the ith term in text k.

The edge weight ω_{ij} represents between words/terms in text, i.e the co-occurrences times. Finally, each text is represent as:

$$(\omega_1, ..., \omega_{|\tau|}, \omega_{11}, ..., \omega_{1|\tau|}, ..., \omega_{|\tau|1}, ..., \omega_{|\tau||\tau|})$$

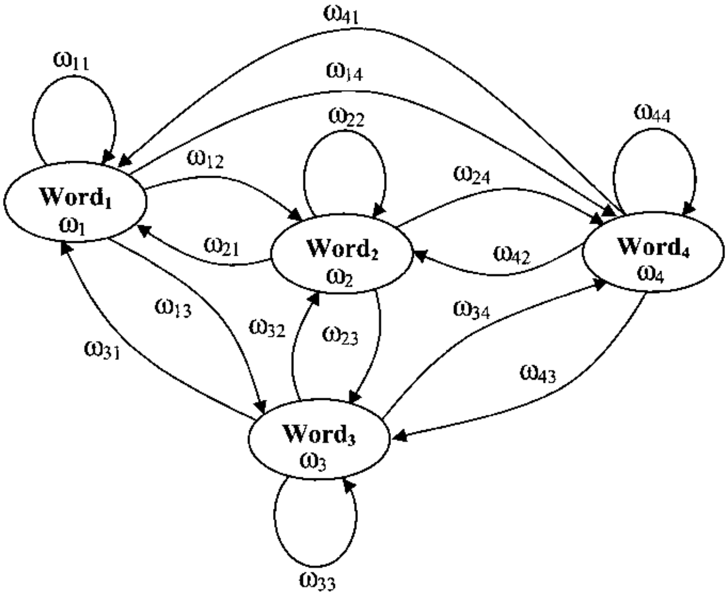


Figure 2.2: Word-based associative relational network [2].

2.2.3 Part-of-Speech

Feldman et al. (2009) used the statistics of POS (part-of-speech) histograms to classify text [10]. By POS tagging, they selected a set of k POS. Then, they used a sliding window ω to count the histogram of the POS set. For example,

$$\phi = \{\{num_{POS_1}, ..., num_{POS_K}\}_1, ..., \{num_{POS_1}, ..., num_{POS_K}\}_n\}$$

Where n is the sliding times, POS_i indicates each POS, and , num_{POS_i} is the frequency of POS_i. With ϕ, $\mu(\phi)$ and $\sigma(\phi)$ are computed, which are the mean and standard deviation of ϕ, respectively. Finally, $[\mu(\phi)\sigma(\phi)]^T$ is used as the feature vector.

2.2.4 N-gram

The traditional n-gram methods use the successive two or several words as the features to classify texts. Sidorov et al. (2012) proposed the syntactic n-grams (sn-grams) to represent texts [40]. The difference between the sn-grams and the traditional n-grams is that the neighbors of words are selected according to the syntactic relations of words, i.e. the dependency tree. Figure 2.3 shows an example of sn-gram.

From this tree, we can extract the following sn-grams:

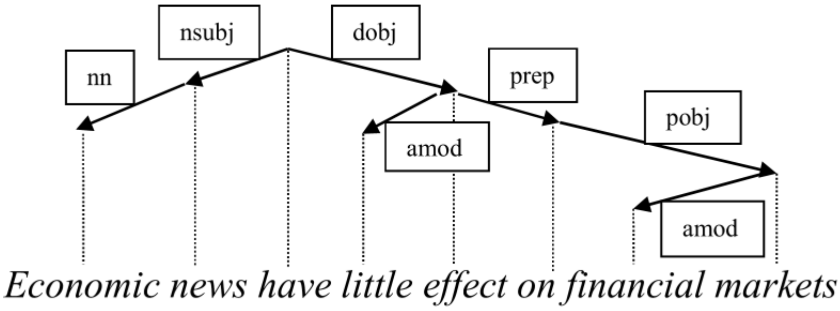


Figure 2.3: An example of sn-grams [40].

- nsubj → nn
- dobj → amod
- dobj → prep → pobj
- prep → pobj → amod
- dobj → prep → pobj → amod

2.2.5 Syntactic and Semantic

Çelik and Güngör (2013) used semantic features such as synonyms, hypernyms, hyponyms, meronyms and topics from WordNet to represent texts [5]. In this method, a word is represented as

$$word_i = \{s_1, s_2, ..., s_m\}$$

Where $word_i$ is the synset of word and s_i is the synset of synonyms, hypernyms, hyponyms, meronyms or topics. And a text is represented as

$$t_i = \{word_1, word_2, ..., word_3\}$$

Finally, a text is represented by the synsets of all words, as

$$t_i = \{s_1, s_2, ..., s_k\}$$

2.2.6 Phrase Pattern

Zhang et al. (2013) extracted the phrase patterns with part-of-speech (POS) to classify texts [53]. This is similar to n-grams, but phrase patterns are the n-grams that allow the gaps between words. Figure 2.4 shows the example of pharse patterns. The extracted result is ({you}, {NEGATIVE POLARITY}, {right, JJ POS})

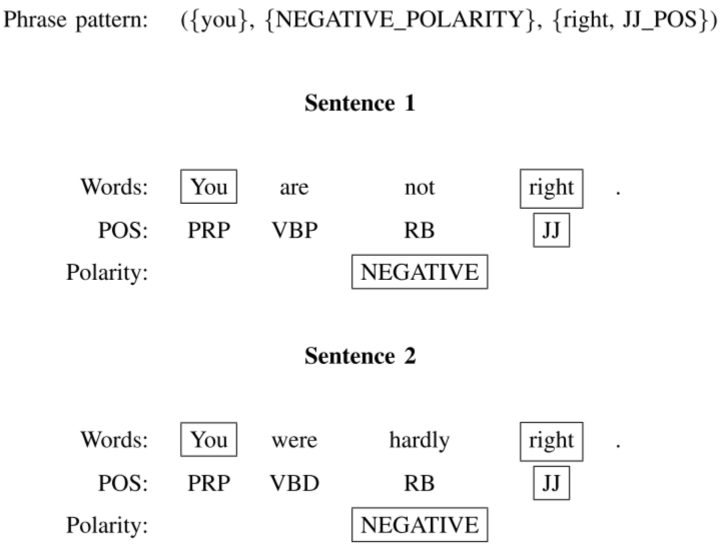

Phrase pattern: ({you}, {NEGATIVE_POLARITY}, {right, JJ_POS})
Sentence 1
Words: You are not right .
POS: PRP VBP RB JJ
Polarity: NEGATIVE
Sentence 2
Words: You were hardly right .
POS: PRP VBD RB JJ
Polarity: NEGATIVE

Figure 2.4: An example of pharse patterns [53].

2.2.7 Feature Transformation

Uysal and Gunal (2014) proposed the genetic algorithm oriented latent semantic features (GALSF), which used filter-based methods and latent semantic indexing (LSI) to select features and represent texts.

2.2.8 Learning Features

Lai et al. (2015) proposed the recurrent convolutional neural networks to learning text representation [20].

Figure 2.5 shows the structure of the recurrent convolutional neural network. In this network, each word is represented by word embedding. The left and right context of the word are computed by the recurrent method. x_i indicates feature vector of the word, its left and right context. Then, x_i is mapped to y_i by convolutional method. With the max-pooling layer, all y_i is converted to a single vector. Finally, the results output from

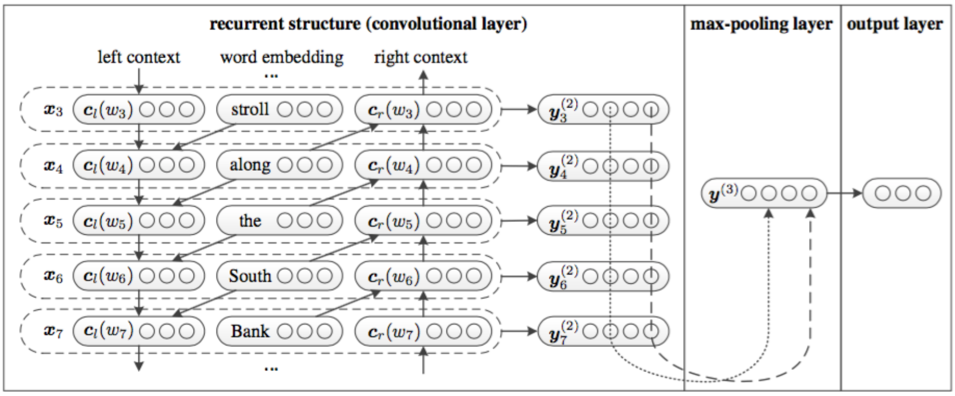


Figure 2.5: The structure of the recurrent convolutional neural network [20].

the max-pooling layer.

2.3 Machine Learning Methods for Text Classification

In the section, we will introduce the machine learning methods for text classification. The machine learning methods and text features are used to train classifiers.

2.3.1 Bayesian models

Naïve Bayes (NB) [17][55] classifier is based on Bayes' theorem, which is widely used in classification task. It assumes that the features are independent from each other. For example, a text is represented by:

$$text = w_1, w_2, ..., w_n$$

This class label for the text is:

$$label(text) = \arg\max_c (P(label = c) \prod_{i=1}^{n} P(w_i | label = c)) \tag{2.5}$$

Where the $P(label = c)$ is the probability of class c, $P(w_i|label = c)$ is the conditional probability of word w_i with class c.

2.3.2 Regression-Based Classifiers

In regression-based method, the Linear Least Squares Fit (LLSF) and Logistic Regression methods are widely used to classify texts.

In LLSF method, the class labels are predicted by $Y = A \cdot X + B$. X is the feature matrix and Y is the class labels. The goal is to learn the A and B with training texts and minimized the value between the true and predicted values, $\sum_{i=1}^{n}(p_i - y_i)^2$.

In Logistic regression (LR) method, the objective function is

$$Y = \frac{\exp(A \cdot X + B)}{1 + \exp(A \cdot X + B)} \tag{2.6}$$

Therefore, LR method limits the range in [0, 1], which is the difference from LLSF.

2.3.3 Support Vector Machines (SVMs)

The main principle of support vector machines (SVMs) is to determine the hyperplane in the search space [1][34]. The hyper plane can best separate the different categories by maximize the distance of hyperplane between all the categories.

Figure 2.6 shows an example of SVM. The examples on the dash lines are the *support vectors* for the two categories, separately, which represent their categories. The solid line is the *hyperplane*. The distance between the hyperplane and the support vectors is called *margin of separation*. In this example, we have the maximum *margin* by using the hyperplane and support vectors to classify these two categories. Our goal is to find this hyperplane.

2.3.4 Centroid

After preprocessing, each text is represented by a numeric vector. In centroid method [34], the centroid of each category is computed by their text vectors:

$$Centroid_k = \frac{\sum_{text \in c_k} text_i}{\sqrt{\sum_{c=i}^{k}(\sum_{text \in c_k} text_i)^2}} \tag{2.7}$$

Where k is the number of categories.

[7] http://scikit-learn.org/stable/modules/svm.html#svm

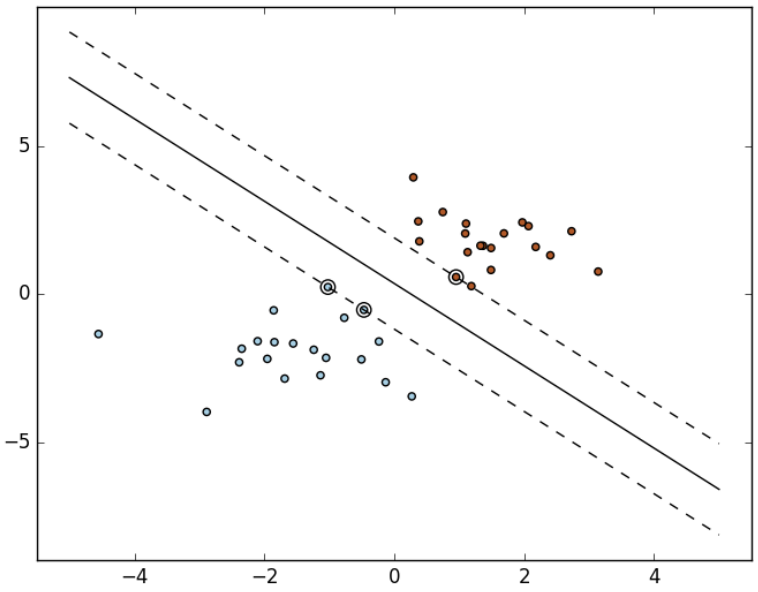


Figure 2.6: An example of SVM.[7]

Finally, the text is assigned is the category which has the maximum similarity with the text.

$$label(text_i) = \arg\max_{k \in C}(text_i \cdot Centroid_k) \tag{2.8}$$

2.3.5 k-Nearest Neighbor (KNN)

K-nearest Neighbor (KNN) is a simple classification algorithm. The idea of KNN is determining the class label for a text according to several other texts (neighbors) which are nearest to it [18][44]. Several similarity algorithms can used to compute the distance between the text and its neighbors, such as Euclidean distance, cosine distance, etc. For an example, there are 5 neighbors for the text X. Three of them belong to class C_1, and Two of them belong to class C_2. Therefore, in this case, X belongs to class C_1.

2.3.6 Clustering

Commonly, clustering methods deal with unlabeled data, splitting examples into several categories. Besides, clustering methods often combine with other methods in text classification task.

For example, Pang et al. (2013) propose an clustering-based KNN method to classify texts. First, they clustered the training te into several clusters for every category. Each cluster has the texts with the same class label. To predict the class label for a new text, the KNN method is used to select the top K neighbors and determine the label for the text.

In Huda et al.'s paper (2017), they clustered training texts into 3 categories which is equal to the number of categories in the corpus. Then, the similarities between the clusters and log texts are added to the feature vectors to classify the log texts.

2.3.7 Active Learning

Active learning (AL) an iterative, semi-supervised learning method that is used to select the most informative examples from unlabelled datasets [14]. Then, the selected examples are labeled by experts. These new labeled examples are used to train the classifier or predict the labels for unlabelled data. AL can reduce the necessary number of training data.

For example, Hu et al. (2016) combined different classifiers to investigate the performance with AL. Figure 2.7 show the flow of AL.

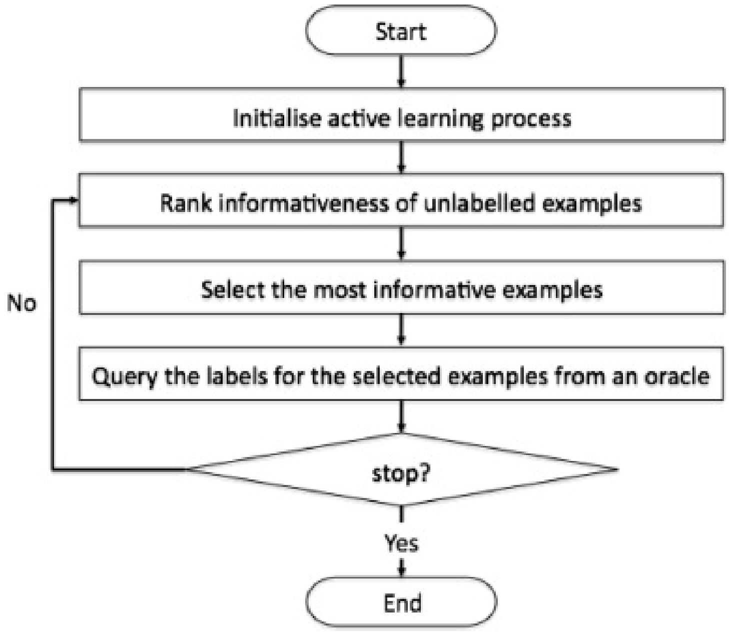

Start
Initialise active learning process
Rank informativeness of unlabelled examples
Select the most informative examples
Query the labels for the selected examples from an oracle
stop?
No
Yes
End

Figure 2.7: A flow-chart of the active learning process [14].

The learning process loops to select examples for labeling. Finally, the process stops with the stopping criterion, for example, the number of selected examples.

2.3.8 Classifier Combination

The classifier combination method uses the results from several classifiers to output the final result. Enríquez et al. (2013) investigated the performance of several combination methods such as voting, Bayesian merging, behavior knowledge space, bagging, stacking, feature sub-spacing and cascading in NLP tasks [9].

In summary, we reviewed the methodologies for text classification in this chapter. We illustrated the framework and flow of text classification. The main steps include text representation, training model and prediction. We also demonstrated the steps and methods in details. For text representation, the preprocessing methods were introduced with the some examples and previous work. For training model, this chapter demonstrated some machine learning methods used for text classification task.

Chapter 3

Related Work

To classify texts, features are very important, and the main difficult is the lack of informative and proper features to represent texts. From this point of view, some researches focused on enriching text representation by combining simple words with some other kinds of features. Moschitti et al. (2004) used POS (part-of-speech), complex nominals, proper nouns, and words senses as the additional features, although they reported that these features did not improve the performance [25]. Alessandro Moschitti (2008), proposed the kernel methods based on words, predicate argument structures (PASs), POS-tag sequences and syntactic parse trees to improve the BOW method for answer classification [24]. Figueiredo et al. (2011) extracted the word co-occurrence as the features, which includes two or more terms [11]. For example, in medical subjects, the pair of {pain, facial} may help determining the document category. However, some pairs do not bring relevant information for the classification, such as {pain,reach} and {pain,beliefs}, which should be filtered out. In Sidorov et al.'s paper, they introduced a concept of syntactic n-grams (sn-grams) which are the n-grams following the links in the syntactic tree [40]. By using the semantic relation in WordNet, Celik and Gungor added semantic features such as synonyms, hypernyms, hyponyms, meronyms and topics into classification process [5]. Rocha et al. extracted the temporal evolution to improve effectiveness for text classification [35]. In the sentiment classification field, the part-of-speech and word-relation based feature sets are designed for classification task [52]. Bravo-Marquez et al. (2014) used many meta-level features to improve the effectiveness, which extracted from several lexicon of sentiment and opinion [4]. In this thesis, similarly to the previous work, we use background knowledge to complement the information for representing texts.

In text classification, the imbalanced corpus is one of the challenges, in which the distribution of texts in categories is not equal. In some extreme case, a small category has only one text, while other categories have a large amount of texts. The two common strategies of classifying imbalanced corpus are over-sampling and under-sampling. The over-sampling is generating new texts to complement the number of texts for small categories. In contrast, the under-sampling is cutting down some texts in large categories to balance the distribution. Chen et al. and Iglesias et al. proposed two over-sampling methods base on HMM and global semantic information respectively [7][16]. Wang et al. presented an under-sampling method by cutting the majority class sample in the boundary region between each categories [51]. By combining both over-sampling and under-sampling method on imbalanced corpus, Qian et al. used the ensemble of methods with voting strategy for classifying [28]. Essentially, all these methods tried to find the proper features in the specific dataset to deal with the imbalanced corpus. However, they ignored the case that there are not enough similar features between training and testing texts in the imbalanced corpus, especially in small categories. In this thesis, we also focus on the imbalanced dataset, and used the background knowledge to complement the information for training and test texts.

In this thesis, the proposed method is inspired by some previous researches.

Ren (2010) proposed the thought of using “Cloud Computing” in natural language processing (NLP) [32]. Because people store their knowledge in their memory. With this idea, if we regards the Could as the memory, we can use background knowledge from the Cloud to assist us in making decision, such as the search engine, the online encyclopedia, etc. In the proposed methods, we use Baidu Baike (an online encyclopedia) and some additional corpora as the background knowledge.

Wang et al. (2009) selected the related articles from Wikipedia for each text, and added these articles to each text to enrich text representation [50]. First, they downloaded the dataset of Wikipedia from `http://download.wikipedia.org`, and used the link structure and articles to make a thesaurus, in which each concept is an article. And then, for each text, they searched the concepts from the text in this thesaurus. The found concepts were the candidate concepts. For example, if the word “apple” is in the text and found in the thesaurus, “apple” is a candidate concept. Because “apple” maybe the company or fruit according to its links in the thesaurus, the next step is the disambiguation which selects the right concepts most related to the text. Finally, the selected the concepts were added to their responding text to enriching text representation and train the classifier.

Rafi et al. (2012) add the titles, redirects, entity types, categories and linked entities from Wikitology to the text, and then Linear SVM was used as the classifier [31]. Wikitology is a knowledge repository which extracts knowledge from Wikipedia in structured/unstructured forms in ontological structure. A title is the topic of a Wikipedia article, such as “Barack Obama”. The redirects are the similar concepts of the topic, such as “President_Barack_Hussain_Obama”. The entity types are the same properties of the topics, such as “Freebase:person”. The category is the is-a-kind-of relation of the topic, such as “United_States_presidential_candidates_2008” is the category for “Barack Obama”. The linked entities are the Persons, Locations, and Organizations along with the topic, such as “Barack Obama” links to “Michelle_Obama” and “University_of_Chicago_Law_School ”.

Torunolu et al. (2013) added Wikipedia article titles, categories and redirects into the feature list to enrich text representation [41]. First, they used the BOW (bag of word) model to represent text. And, they got a feature list which has all words from texts. By searching each text, if any wiki concepts are found, the categories and redirects of the concepts are added to the feature list. The maximum length of wiki concepts is limited to three words. With the new feature list, they count the frequency of each feature to enrich text representation.

The methods mentioned above used the Wikipedia as the background knowledge to assist in classifying texts. The difference between these methods and ours is that we focus on the similarities between texts and the background knowledge, rather than adding the additional content to texts. And, we focus on Chinese language which has its unique properties. For example, the English characters are from a to z, which are letters without any meaning, but each Chinese character has meaning itself as mentioned above.

Focusing on the distribution of words in different categories, Ren and Sohrab (2013) proposed TF.IDF.ICF and TF.IDF.ICS$_\delta$F methods [34]. These two methods are used to weight terms for representing texts. The former is class-frequency(CF)-based category mapping of each term. The later is class-space-density-based category mapping of each term.

Commonly, the TF.IDF algorithm weights the terms for texts. The TF.IDF.ICF is TF.IDF multiplying by ICF. For each term, the TF.IDF.ICF weight is computed by equation 3.1:

$$W_{TF.IDF.ICF}(t_i, d_j, c_k) = tf_{(t_i,d_j)} \times (1 + \log \frac{D}{d(t_i)}) \times (1 + \log \frac{C}{c(t_i)}) \tag{3.1}$$

where t_i, d_j and c_k indicate the ith term of text j in category k, $tf_{(t_i,d_j)}$ is the term frequency, D is the total number of texts, $d(t_i)$ is the number of the texts that include term i, C is the total number of categories, $c(t_i)$ is the number of categories that include term i, $\frac{C}{c(t_i)}$ is the ICF of term t_i. And the normalization is:

$$W^{norm}_{TF.IDF.ICF}(t_i, d_j, c_k) = \frac{W_{TF.IDF.ICF}(t_i, d_j, c_k)}{\sqrt{\sum_{t_i \in d_j; t_i \in c_k} [W_{TF.IDF.ICF}(t_i, d_j, c_k)]^2}} \tag{3.2}$$

The TF.IDF.ICS$_\delta$F is TF.IDF multiplying by ICS$_\delta$F. First, the class density C_δ is the rate of the texts that include the term t_i in the category c_k. For each term, the equation is,

$$C_\delta(t_i) = \frac{n_{c_k}(t_i)}{N_{c_k}}$$

where $n_{c_k}(t_i)$ is the number of the texts that include term t_i in the category c_k, and N_{c_k} is the total number of the texts in the category c_k.

And, to compute the class space density (CS_δ), the equation is,

$$CS_\delta(t_i) = \sum_{c_k} C_\delta(t_i)$$

Therefore, for each term, the inverse class space density frequency is computed by following,

$$ICS_\delta F(t_i) = \log(\frac{C}{CS_\delta(t_i)})$$

where C is the total number of categories.

Finally, the TF.IDF.ICS$_\delta$F weight for each term is:

$$W_{TF.IDF.ICS_\delta F}(t_i, d_j, c_k) = tf_{(t_i,d_j)} \times (1 + \log \frac{D}{d(t_i)}) \times (1 + \log \frac{C}{CS_\delta(t_i)}) \tag{3.3}$$

And the normalization is:

$$W^{norm}_{TF.IDF.ICS_\delta F}(t_i, d_j, c_k) = \frac{W_{TF.IDF.ICS_\delta F}(t_i, d_j, c_k)}{\sqrt{\sum_{t_i \in d_j; t_i \in c_k} [W_{TF.IDF.ICS_\delta F}(t_i, d_j, c_k)]^2}} \tag{3.4}$$

Inspiring by their idea, we propose a method which use the weights of terms on category level to select the keywords for each category.

Chapter 4

Extracting Background Knowledge for Text Classification

In this thesis, we propose the methods based on background knowledge to enrich text representation for text classification task. Two kinds of background knowledge are used in this study, the information from Baidu Baike and character co-occurrence.

In this chapter, we first introduce the traditional method for representing and classifying texts in section 4.1. Section 4.2 presents the method based on Baidu Baike to enrich text representation. Section 4.3 presents the method based on character co-occurrence to enrich text representation. Section 4.4 presents the measure for evaluating the performance of the proposed methods. Section 4.5 introduces the tools used in the proposed methods, including the NLP tools and machine learning tools.

4.1 Traditional Method for Text Classification

Commonly, texts are represented by using Vector Space Model (VSM). The index of the vector represents a sequence of features. The length of the vector is the total number of features from the corpus. The vector consists of numeric values, and each value represents the weight for the term. Commonly, the features are words. There are also some other kinds of features, such as n-gram, phrase, syntactic structure. VSM in this study, converting the texts to the vectors of numeric values. In traditional method, the features are the words and weighted by TF-IDF algorithm. After representing texts as vectors, training texts could be convert to a feature matrix whose rows are the texts, columns are the features, and each value in this matrix is the weight of the feature.

The workflow of the traditional method is:

1. The corpus is preprocessed by some NLP (natural language processing) technologies, such as word segment, and POS (part of speech) tagging

2. Features are extracted from the training texts to build the feature list.

3. The training texts are represented by the vectors of weighted features with the feature list.

4. Some machine learning methods use the training texts to train classification model.

5. Test texts are also represented by the vectors of weighted features with the same feature list.

6. The model predicts the class labels for the test texts.

7. Finally, it is evaluating the performance for the model .

4.2 Background Knowledge from Baidu Baike

In this section, we introduce enriching text representation based on Baidu Baike. Baidu Baike is a Chinese online encyclopedia which has a lot of articles about many different concepts, such as persons, objects, events, abstracts, etc.

The main idea of this method is that more information could improve the effectiveness for text classification. The motivation is from dealing with imbalanced corpus. In this case, the large categories have much more texts than the small categories. Moreover, sometimes these is not any similar information between the training and test texts, because these is not enough numbers of texts in the categories.

In this method, Baidu Baike is used as the knowledge base to enrich information for Chinese texts. The information from knowledge base is treated as background knowledge. The information from the training and testing texts in a corpus is regared as a special case.

Section 4.2.1 illustrates the proposed method with the background knowledge from Baidu Baike. Section 4.2.2 presents the details about implementing the proposed method.

4.2.1 Overview: Using Baidu Baike

In this method, Baidu Baike is used as the knowledge base to obtain some concepts and general knowledge. In other words, the articles from Baidu Baike are used as the background knowledge in this method.

Figure 4.1 illustrates the workflow of classifying texts based on Baidu Baike. In Figure 4.1, there are a lot of steps. In the middle of the figure, these steps are the difference between the proposed method and the traditional method. On the left, they are the training process. On the right, they are the predicting process. Every steps are introduce as following:

- The steps at the top: these three steps represent the experimental data.

  Training text The training texts are from the corpus and includes numbers of texts in each category. They are used for training classification model.

  Label This represents the class labels of the training texts.

  Test text The test texts are also from the corpus and includes numbers of texts in each category. They are used to evaluate the performance of the method.

- The steps in the middle:

  Words represent Categories In this step, we select several candidate words that are most related to their categories by using CTF-ICF algorithm.

  Baidu Search Engine Baidu Search Engine is used to select top N keywords for their categories. Each candidate word and their category's name are posted to the search engine, and these candidate words are ranked by the number of search results.

  Top N words of Categories The top N keywords are selected by ranking the number of search results. We assume that it gets more search results, the candidate word is more related to its category.

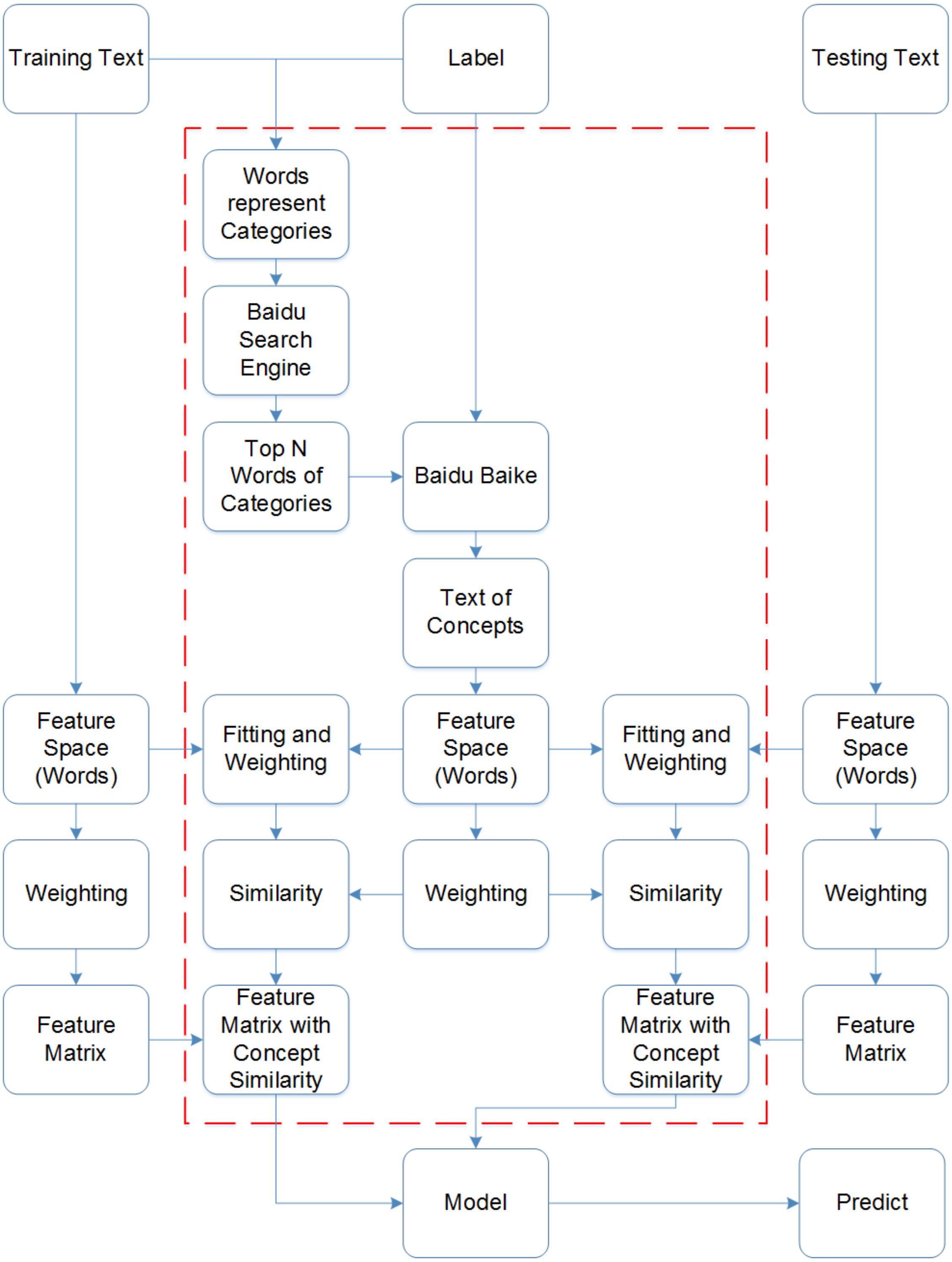


Figure 4.1: Workflow: enriching text representation with Baidu Baike

Baidu Baike With the top N keywords for each category, we post them to Baidu Baike.

Text of Concepts The articles of the top N keywords are download from Baidu Baike and assign their class labels to them. These articles are regarded as the background knowledge.

Feature Space (words) In this step, we extract the features from the background knowledge. These features are the words in the articles, called as background features.

Fitting and Weighting This step uses the background features to represent the training and test texts by filtering out the words that are not in the background features from the texts.

Weighting This step converts the articles in the background knowledge to the vectors of numeric values.

Similarity Computing the similarity between each text and each article. For example, if there are 5 articles in the background knowledge, we will get 5 similarities for each text.

Feature Matrix with Concept similarity Enriching text representation by adding the similarities to the feature vectors of texts.

- The steps on the left:

Feature Space (Words) Extracting the features from the training texts. The features are the words in the training texts

Weighting Weighting the features and representing the texts as the vectors of numeric values.

Feature Matrix In the feature matrix, the rows are the texts, the columns are the features, the values are the weight.

- The steps on the right:

Feature Space (Words) Extracting the features from the test texts. The features are the words in the training texts

Weighting Weighting the features and representing the texts as the vectors of numeric values.

Feature Matrix In the feature matrix, the rows are the texts, the columns are the features, the values are the weight.

- The steps at the bottom:

Model Training the classification model with the feature matrix which includes the similarities between the texts and the background knowledge.

Predict Predicting class labels for the test texts with the classification model.

4.2.2 Enriching text representation with Baidu Baike

In this method, the articles of the category names and top N keywords are downloaded from Baidu Baike. Each article is saved to a text as the background knowledge. The words are used as the features for the texts. To convert a text to a vector, the TF-IDF algorithm is used to weight the features.

To select the top N keywords for each category, we use CTF-ICF algorithm to find out the candidate words for each category from the training texts. CTF-ICF algorithm is similar to TF-IDF algorithm, but it works on the category level. The CTF-ICF algorithm is:

$$W_{CTF\cdot ICF}(t_i, c_k) = (1 + \log ctf(t_i, c_k)) \times (1 + \log \frac{|C|}{cf(t_i)}) \tag{4.1}$$

where (t_i, c_k) indicates the feature t_i in the category c_k, $ctf(t_i, c_k)$ is the number of the feature in the category c_k, $|C|$ is the total number of all categories, $cf(t_i)$ is the number of categories that include t_i. $\frac{|C|}{cf(t_i)}$ is the ICF that is the same as [34]. The result of $\log ctf(t_i, c_k)$ is 0, when $ctf(t_i, c_k) = 1$. Therefore, we use $1 + \log ctf(t_i, c_k)$ to make the result not equal to 0, which can make $W_{CTF\cdot ICF}(t_i, c_k)$ not equal to 0. For the same reason, we use $1 + \log \frac{|C|}{cf(t_i)}$ instead of $\log \frac{|C|}{cf(t_i)}$.

And then, each candidate word and its category name are posted to the search engine, and get the number of search results. Finally, the top N words, which are most related to their categories, are selected by ranking the number of search results.

Specially, our experiments only use the category names at the beginning, and the results show the improvement to macro F1 score But the precision a little decreases by Linear SVM classifier. To improve both macro F1 score and precision, the top N keywords for each category are included in the subsequent experiments.

Without the background knowledge, the TF-IDF algorthim is used to convert the training and test texts to the vectors of numeric values. The features are the words from the training texts. For the test texts, the words, which are not in the training texts, are filtered out.

To complement the information from the background knowledge to the texts, we filter out the words, which are not in the background knowledge, from the training and test texts. The filtered training and test texts are represented by using TF-IDF values. The similarity between each text and each article in the background knowledge is computed by similarity algorithm. For each text, we get a similarity vector, and also have the feature vector which are generate without the background knowledge.

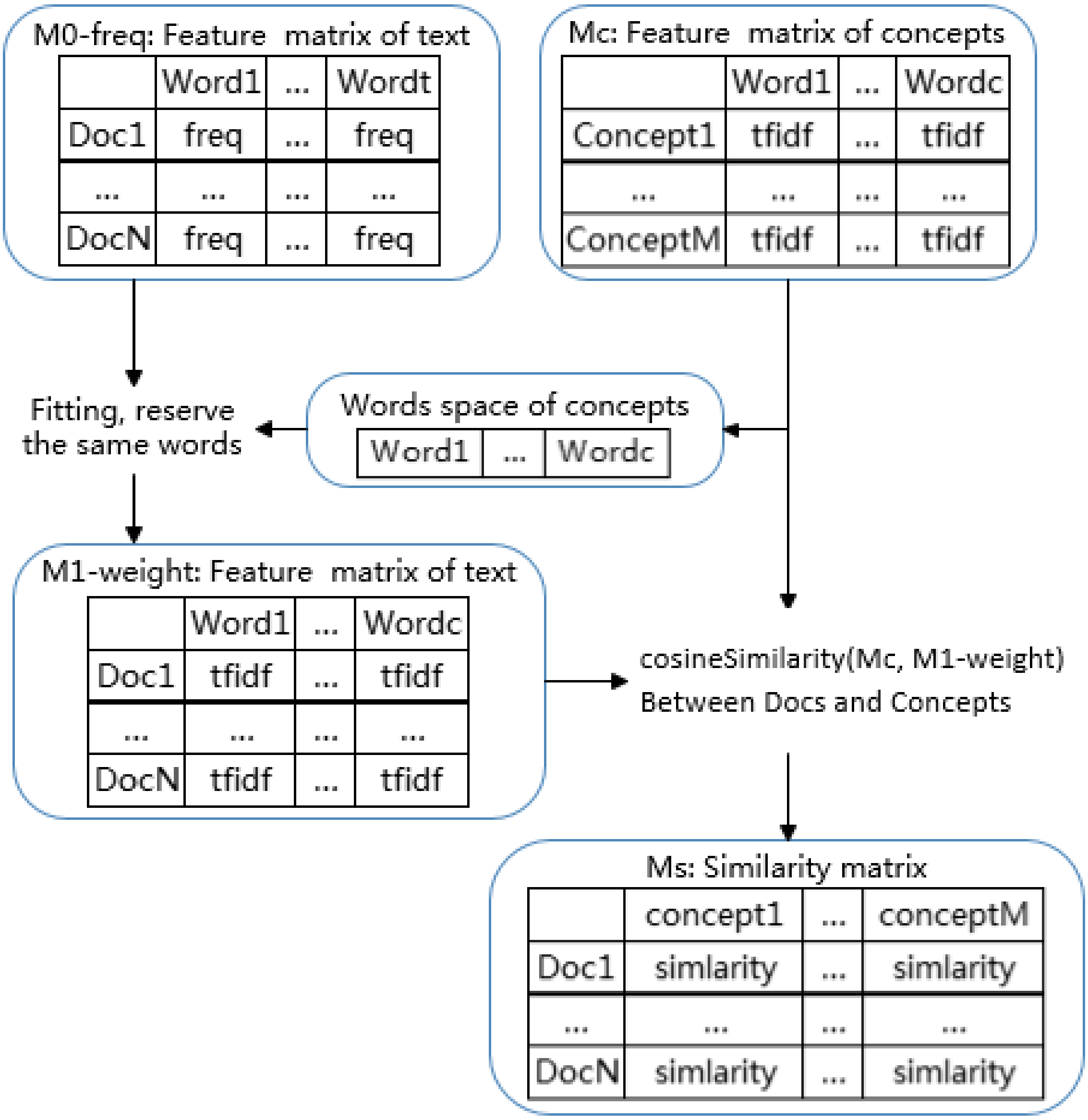
To compute the similarities, cosine similarity algorithm is used,because the texts have already represented by the numeric vectors. The equation of cosine similarity is defined as following:

$$CosineSimilarity = \frac{u \cdot v}{||u||_2||v||_2} = \frac{\sum_{i=i}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}} \tag{4.2}$$

where u and v are two numeric vector, u_i and v_i are the items of vector u and v respectively.

Figure 4.2 illustrates the process that computes the similarities matrix from the training/test texts and the articles in the background knowledge. In each matrix, the rows represent the texts and the columns represent the features. Other details are introduced as following:

- *freq* is the word frequency in the text.
- *tfidf* is the TF-IDF value.
- *concepts* represent the articles from Baidu Baike.
- In the training texts, we may have t words.
- In the *concepts*, we may have c words.
- The vocabularies of the training texts and *concepts* are different.
- The number of *concepts* is dependent on the number of the top N keywords.



M0-freq: Feature matrix of text

	Word1	...	Wordt
Doc1	freq	...	freq
...	...	...	...
DocN	freq	...	freq

Mc: Feature matrix of concepts

	Word1	...	Wordc
Concept1	tfidf	...	tfidf
...	...	...	...
ConceptM	tfidf	...	tfidf

Words space of concepts

Word1	...	Wordc

M1-weight: Feature matrix of text

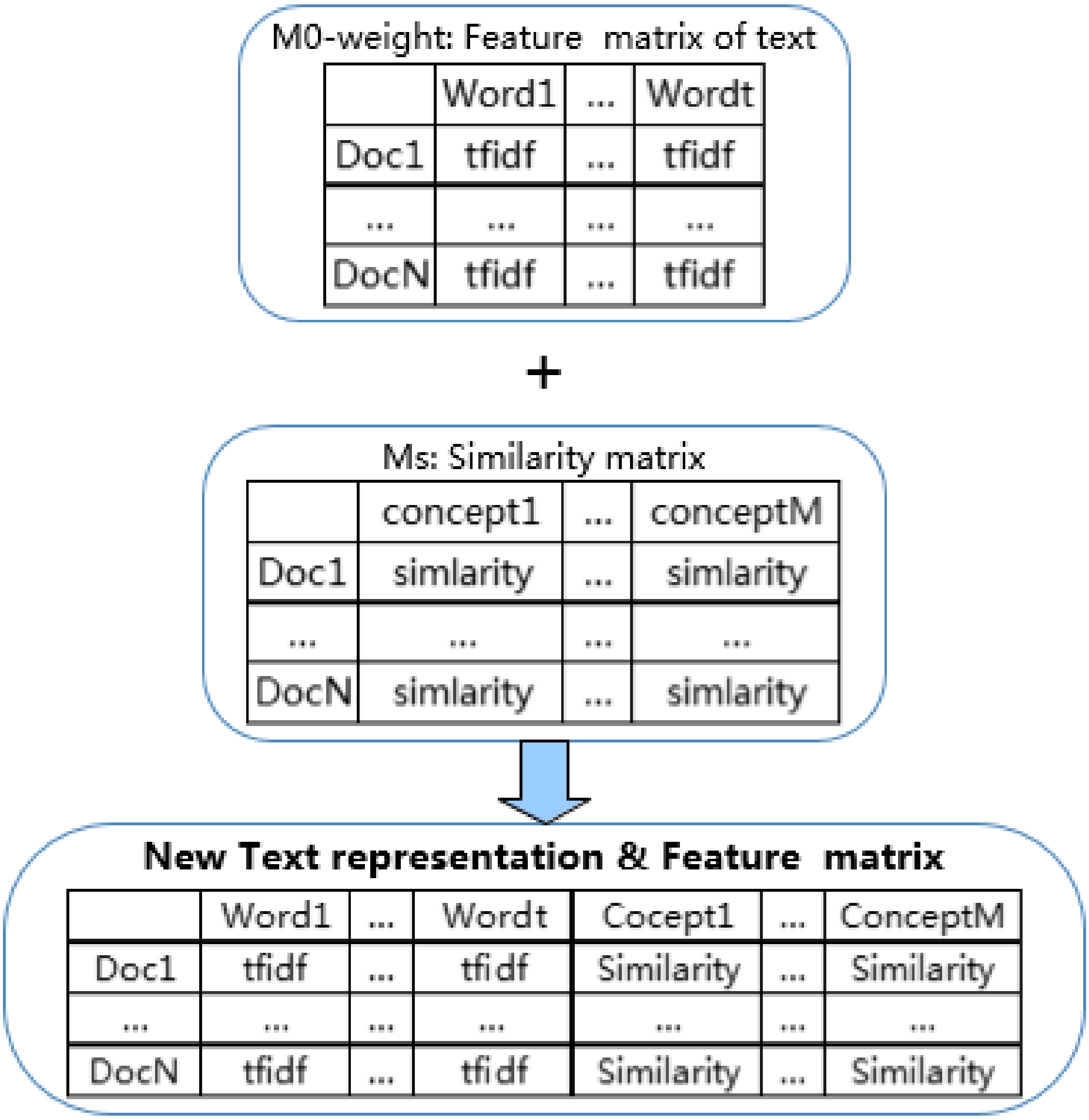
	Word1	...	Wordc
Doc1	tfidf	...	tfidf
...	...	...	...
DocN	tfidf	...	tfidf

Ms: Similarity matrix

	concept1	...	conceptM
Doc1	simlarity	...	simlarity
...	...	...	...
DocN	simlarity	...	simlarity

Figure 4.2: The similarity matrix based on the background knowledge. M_{0-freq} is the feature matrix with word frequency for the training/testing texts. M_c is the feature matrix with TF-IDF values for the articles from Baidu Baike. The words from the article are regarded as the background features. $M_{1-weight}$ is the feature matrix with TF-IDF values for the training/testing texts, in which only the words in the background features are reserved. M_s is the feature matrix with similarities between the texts and the articles from Baidu Baike.

Finally, we combine the similarity vector and the feature vector to represent the text. Figure 4.3 shows this process. With the new feature matrix, the SVM method is used to train classification model and predict the class labels for the test texts.



M0-weight: Feature matrix of text

	Word1	...	Wordt
Doc1	tfidf	...	tfidf
...	...	...	...
DocN	tfidf	...	tfidf

+

Ms: Similarity matrix

	concept1	...	conceptM
Doc1	simlarity	...	simlarity
...	...	...	...
DocN	simlarity	...	simlarity

New Text representation & Feature matrix

	Word1	...	Wordt	Cocept1	...	ConceptM
Doc1	tfidf	...	tfidf	Similarity	...	Similarity
...	...	...	...	...	...	...
DocN	tfidf	...	tfidf	Similarity	...	Similarity

Figure 4.3: Generate the new feature matrix to represent the texts. $M_{0-weight}$ is the matrix with TF-IDF values of words for the training/testing texts. M_s is the matrix with similarities between the texts and the articles from Baidu Baike.

Specially, only nouns and verbs are used as the top N keywords for their categories. Other words are filtered out, such as number, punctuation, and other symbols, as well as excluding the category names because they have been already selected as the keywords. Therefore, the articles of the top N keywords and the category names are downloaded from

Baidu Baike. When look up them in Baidu Baike, some *concepts* have the list of several sub-concepts. Commonly, all of the *sub-concepts* are downloaded into one text. However, some *concepts* do not have the list of *sub-concepts*. In this case, only the first *sub-concept* in the search results is downloaded.

4.3 Background Knowledge From Character Co-occurrence

In this section, we introduce enriching text representation based on character co-occurrence. We extract to the information of character co-occurrence from the additional corpus which is independent from the training and test texts. For example:

In Chinese: 吃苹果。喝苹果汁。

In English: *Eat apples. Drink apple juice.*

In the Chinese text, there are two sentences. By counting the character co-occurrence in each sentence, the results are:

- 吃-苹: 1
- 吃-果: 1
- 苹-果: 2
- 喝-苹: 1
- 喝-果: 1
- 喝-汁: 1
- 苹-汁: 1
- 果-汁: 1

The number indicates the frequency of the co-occurrence. The character “苹” can be represented as:

	吃	苹	果	喝	汁
苹	1	2	2	1	1

Table 4.1: An example for the character co-occurrence in Chinese. The number of 苹-苹 represents the frequency of the character 苹.

Two key ideas support this method. First, each Chinese character has its own meaning. Second, in different topics, the character co-occurrence is different in the texts. Because the number of common characters in Chinese is large, around 20,000. If we count the character co-occurrence for each character, the computation will be $20,000^2$ times. Therefore, to reduce the computation, some key characters are selected, which are the most related to their categories. The character co-occurrence between the key characters and other characters are counted for each text and the additional corpus. For each text, the key characters can also be represented as the vectors with the frequency of the character co-occurrence in itself. For the additional corpus, the key characters can be represented as the vectors with the frequency of the character co-occurrence in itself. Because the content is different in the texts and the additional corpus, the vectors are different for the same key characters. The similarities between the key characters in the texts and the key characters in the additional corpus are used to enrich the text representation.

The motivation is also from dealing the imbalance corpus. In this case, the large categories have much more texts than the small categories. Moreover, sometimes these is not any similar information between training and test texts, because these is not enough numbers of texts in the categories.

In this method, the additional corpus is used as the knowledge base to enrich information of Chinese texts. The information from knowledge base is treated as background knowledge. The information from training and test texts in a specific corpus is a special case.

Section 4.3.1 illustrates the proposed method using character co-occurrence as the background knowledge. Section 4.3.2 presents the details about implementing the proposed method.

4.3.1 Overview: Using Character Co-occurrence

In this method, the additional corpus is used as the knowledge base to count the character co-occurrence. In other words, the character co-occurrence from the additional corpus is used as the background knowledge.

Figure 4.4 illustrates the workflow of enriching text representation based on character co-occurrence. In Figure 4.4, there are a lot of steps. In the middle of the figure, these

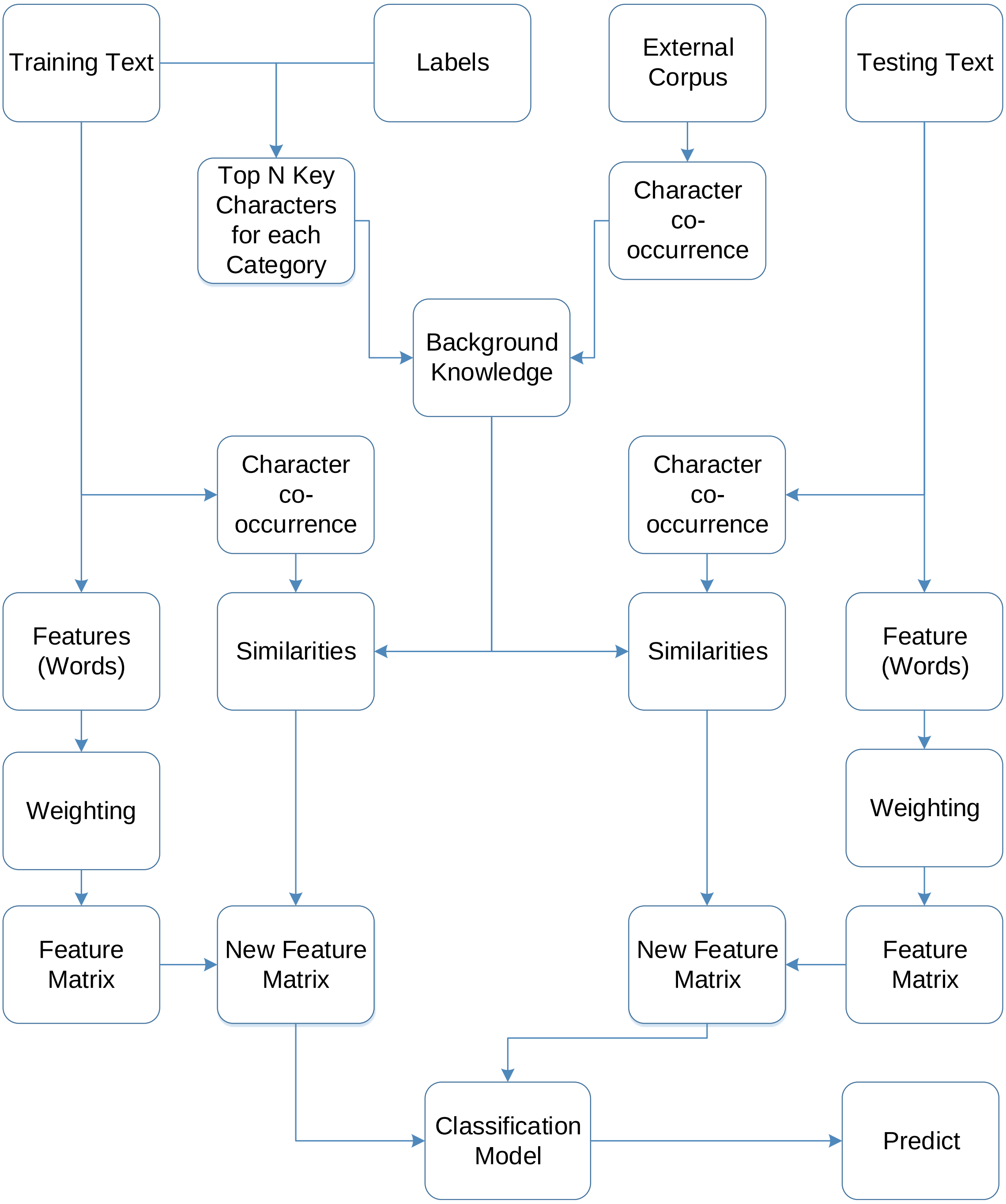


Figure 4.4: Workflow: enriching text representation with character co-occurrence

steps are the difference between the proposed method and the traditional method. On the left, they are the training process. On the right, they are the predicting process.

Every steps are introduce as following:

- The steps at the top: these four steps represent the experimental data.

  Training text The training texts are from the corpus and includes numbers of texts in each category. They are used for training classification model.

  Labels This represents the class labels of the training texts.

  External Corpus The external is used to count the character co-occurrence which is regarded as the background knowledge.

  Test text The test texts are also from the corpus and includes numbers of texts in each category. They are used to evaluate the performance of the method.

- The steps in the middle:

  Top N key characters for each category In this step, we select top N characters which are the most related to their categories.

  Background Knowledge To reduce the computation, only the top N key characters are selected to compute the character co-occurrenceas, which is used as the background knowledge.

  Similarities Computing the similarities of the key characters between each text and the background knowledge. For example, if there are 5 key characters, we will get 5 similarities for each text.

  New Feature Matrix Enriching text representation by adding the similarities to the feature vectors of the texts. In the feature matrix, the rows are the texts, the columns are the features, the values are the weight and the similarities.

- The steps on the left:

  Feature Space (Words) Extracting the features from the training texts. The features are the words from the training texts.

  Weighting Weighting the features and representing texts as the vectors of numeric values.

Feature Matrix In the feature matrix, the rows are the texts, the columns are the features, the values are the weight.

- The steps on the right:

Feature Space (Words) Extracting the features from the test texts. The features are the words from the training texts.

Weighting Weighting the features and representing texts as the vectors of numeric values.

Feature Matrix In the feature matrix, the rows are the texts, the columns are the features, the values are the weight.

- The steps at the bottom:

Classification Model Training the classification model with the feature matrix which includes the similarities between the texts and the background knowledge.

Predict Predicting the class labels for the test texts with the classification model.

4.3.2 Enriching text representation with Character Co-occurrence

In this method, the additional corpus is downloaded from Internet. With this corpus, we count the character co-occurrence which are used as the background knowledge.

The training and test texts are represented by the BoW model, and each feature is weighted by the TF-IDF algorithm. Each text is converted to a vector of numeric values.

In Figure 2, the steps in the middle illustrate the process of using the additional corpus to obtain the character co-occurrence information, which is used as the background knowledge.

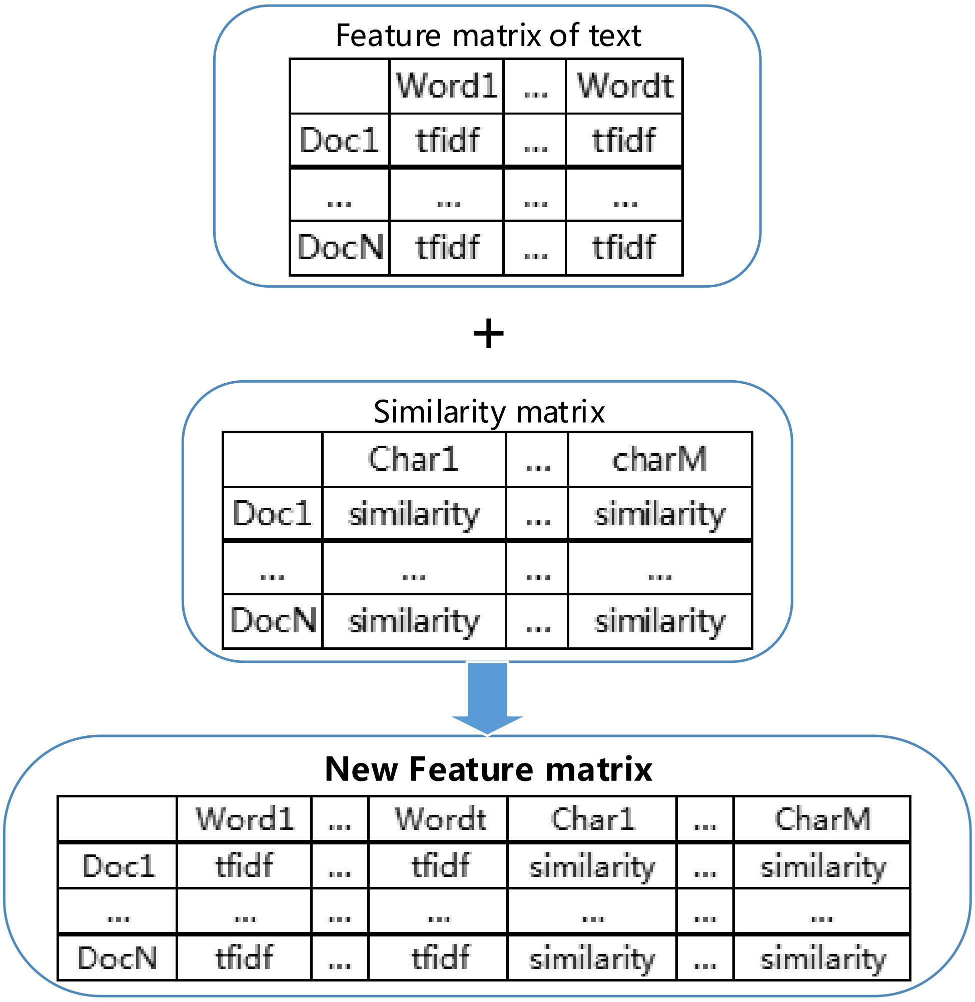
First, the top N key characters are selected according to the texts and their class labels, in which each character is regarded as having the discriminative ability corresponding to its category. Second, the additional corpus is used to obtain the character co-occurrence frequencies which could constitute a character co-occurrence matrix. Only the top N key characters are extracted from the co-occurrence matrix as the background knowledge. Third, each text computes its own character co-occurrence frequencies of the top N key characters. The similarities of character co-occurrence frequencies between the background knowledge and each text are used as the new features. Finally, these new features are added to the feature matrix to train classification model. Note that, the similarities of character co-occurrence frequencies are also added to the test texts.

To select the top N key characters for each category, only nouns are reserved in the training texts at first; And Then, the characters are weighted in each category and ranked by their weight. Finally, the top N key characters are selected.

To obtain the character co-occurrence frequencies, only the Chinese characters are reserved in the texts, others are replaced by \n (Enter character). More than one successive \n are replaced by only one \n character, so that the texts are rearranged as one sentence on each line. More than one successive space character (white space or \t character) are replaced by only one white space character. By processing each sentence in the texts, we count the co-occurrence frequencies between two characters in the sentences. And then, these characters and co-occurrence frequencies could form a character co-occurrence matrix, in which the rows and columns represent the characters, the values represent the co-occurrence frequencies of two characters indexed by the row and column. The distance between two characters are also considered when computing the co-occurrence matrix. For examples, the distance is equal to 1, which means two characters are adjacent. The distance is equal to 2, which means two characters are separated by one character. With different distance, we compute the different character co-occurrence matrices separately. To decide the number of the groups of distance, we compute the average length of the sentences in the additional corpora. The average length of the sentences is around 7 to 8 character. Therefore, we use 6 characters as the maximum distance between two characters in the sentence.

For preprocessing the background knowledge, the character co-occurrence matrix is computed with the additional corpus. Specially, after preprocessing the additional corpus, the characters and their indexes are same in both rows and columns. Therefore, the diagonal represents the frequencies of characters. With the purpose of only using co-occurrence frequencies between different characters, the diagonal of the co-occurrence matrix is set to 0. And then, each row is divided by the sum of values of itself to obtain the co-occurrence rates. Finally, to represent the background knowledge, only the rows which represent the top N key characters are selected.

For preprocessing the training and test texts, the rows of the co-occurrence matrix are the top N key characters, the columns are the same as the columns of the background knowledge, and the values are the co-occurrence rates which depend on the content of each text itself. To compute the similarities, cosine similarity algorithm (equation 4.2) is used, because the co-occurrence frequencies of characters have already represented by the numeric vectors.



Feature matrix of text

	Word1	...	Wordt
Doc1	tfidf	...	tfidf
...	...	...	...
DocN	tfidf	...	tfidf

+

Similarity matrix

	Char1	...	charM
Doc1	similarity	...	similarity
...	...	...	...
DocN	similarity	...	similarity

New Feature matrix

	Word1	...	Wordt	Char1	...	CharM
Doc1	tfidf	...	tfidf	similarity	...	similarity
...	...	...	...	...	...	...
DocN	tfidf	...	tfidf	similarity	...	similarity

Figure 4.5: Combine the feature matrices of TF-IDF values and similarities to represent the texts.

To add the co-occurrence feature to the traditional feature space, we combine the two matrices by columns as showed in Figure 4.5. Other details are introduced as following:

- *tfidf* is the TF-IDF value.
- *similarity* is the similarities between the texts and the background knowledge of the character co-occurrence.
- *Wordt* indicates that we may have t words from the training texts.
- *CharM* indicates that we may have selected the top M words from the additional corpus .

Finally, the SVM (support vector machine) classifier is used to train classification model.

4.4 Evaluation

The evaluation is an important step which tells us the performance of the proposed methods. For evaluating the performance, the standard methods are using precision, recall, F_1 measure. The precision $P(C_k)$, recall $R(C_k)$, F_1 measure $F_1(C_k)$ are defined as follows [34]:

$$P(C_k) = \frac{TP(C_k)}{TP(C_k) + FP(C_k)} \tag{4.3}$$

$$R(C_k) = \frac{TP(C_k)}{TP(C_k) + FN(C_k)} \tag{4.4}$$

$$F_1(C_k) = \frac{2 \cdot P(C_k) \cdot R(C_k)}{P(C_k) + R(C_k)} \tag{4.5}$$

C_k is the target category. $TP(C_k)$ is the number of test texts correctly classified to the category C_k. $FP(C_k)$ is the number of test texts incorrectly classified to the category. $FN(C_k)$ is the number of test texts wrongly denied to the category. To obtain the overall performance, macro F_1 value is used to measure the proposed method. The macro-average of precision (P^M), recall (R^M), and the F_1^M measure are computed as [34]:

$$P^M = \frac{1}{m}\sum_{k=1}^{m} P(C_k) \tag{4.6}$$

$$R^M = \frac{1}{m}\sum_{k=1}^{m} R(C_k) \tag{4.7}$$

$$F_1^M = \frac{1}{m}\sum_{k=1}^{m} F_1(C_k) \tag{4.8}$$

4.5 Tools

In this section, we introduce the tools used in our methods.

Baidu Search Engine Baidu Search Engine[1] is a search engine in Chinese language for websites. In this thesis, Baidu Search Engine is used to obtain the most related keywords which represent their categories.

Baidu Baike Baidu Baike[2] is a Chinese language collaborative Web-based encyclopedia provided by Baidu. In this thesis, the articles of keywords and category name are downloaded from Baidu Baike.[3]

Stanford Word Segment Tokenization of raw text is a standard pre-processing step for many NLP tasks. For English, tokenization usually involves punctuation splitting and separation of some affixes like possessives. Other languages require more extensive token pre-processing, which is usually called segmentation. The Stanford Word Segmenter[4] (SWS) currently supports Arabic and Chinese. The provided segmentation schemes have been found to work well for a variety of applications. In this thesis, SWS is used to segment the text in Chinese into words [6].

Stanford Postagger A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other

[1] http://www.baidu.com

[2] http://baike.baidu.com

[3] http://en.wikipedia.org/wiki/Baidu_Baike

[4] http://nlp.stanford.edu/software/segmenter.shtml

token), such as noun, verb, adjective, etc. In this thesis, Stanford postagger[5] is used to select words with specific POS, such as NN (noun) [43, 42].

Scikit-learn Scikit-learn[6] (formerly scikits.learn) is an open source machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, logistic regression, naive Bayes, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.[7] In this thesis, Scikit-learn provide several function in experiments, such as TF-IDF, Cosine similarity, and Linear SVM classifier. To evaluate the results, we also use the function to compute the precision, recall, and F1-score value and cross validation provided by Scikit-learn.

Numpy and Scipy Numpy and Scipy are two open source libraries for Python programming language which support matrix computation and provide cosine similarity function. Specially, the cosine similarity function (scipy.spatial.distance.cosine(u, v)) in the scipy library is defined as following:

$$ScipyCosineSimilarity = 1 - \frac{u \cdot v}{||u||_2 ||v||_2} \tag{4.9}$$

In this thesis, we use $1 - scipy.spatial.distance.cosine(u, v)$ to compute cosine similarity.

[5] http://nlp.stanford.edu/software/tagger.shtml

[6] http://www.scikit-learn.org

[7] http://en.wikipedia.org/wiki/Scikit-learn

Chapter 5

Evaluation

In this chapter, we evaluate the performance of the proposed methods with some groups of experiments. Section 5.1 presents the experimental results and the discussion for enrich text representation by using Baidu Baike as the background knowledge. Section 5.2 presents the experimental results and the discussion for enrich text representation by using character co-occurrence as the background knowledge. Section 5.3 summarizes these two methods in this chapter.

5.1 Classifying texts with Baidu Baike

The performance of the method based on Baidu Baike is measured with the FUDAN University classification corpus[1,2]. This corpus is from Chinese natural language processing group in Department of Computer Information and Technology in Fudan University.

Section 5.1.1 introduces the dataset used in the experiments. Section 5.1.2 introduces the baseline method. Section 5.1.3 presents the experimental setting. Finally, the experimental results are showed and discussed in section 5.1.4 and section 5.1.5.

5.1.1 Dataset

In the experiments, FUDAN University text classification corpus is used to train classification model. This corpus is an imbalanced dataset, including 19637 texts within 20 categories, in which the different categories have the different numbers of texts. Table

[1] http://www.datatang.com/data/44139

[2] http://www.datatang.com/data/43543

5.1 shows the categories of FUDAN corpus. There are two subsets in FUDAN corpus, including 9084 training texts and 9833 test texts separately.

Figure 5.1 show the the distribution of texts in the categories in FUDAN corpus. And Figure 5.2 and 5.3 shows the distribution of the training and test texts. The x-axis indicates the categories. The y-axis indicates the number of texts. At the top of each bar, it is the number of texts in the category.

C11-Space	C15-Energy	C16-Electronics	C17-Communication
C19-Computer	C23-Mine	C29-Transport	C31-Enviornment
C32-Agriculture	C34-Economy	C35-Law	C36-Medical
C37-Military	C38-Politics	C39-Sports	C3-Art
C4-Literature	C5-Education	C6-Philosophy	C7-History

Table 5.1: The categories of FUDAN corpus.

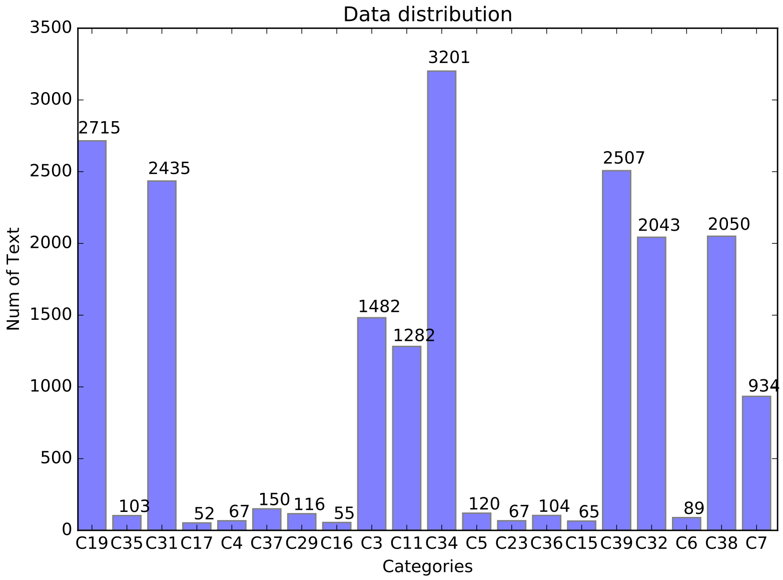


Figure 5.1: The distribution of text in FUDAN corpus.

5.1.2 Baseline

To prove the effectiveness of the proposed method, the INNTC (improved KNN algorithm for text categorization) [18] method is used as the baseline, because they also focused on Chinese text classification and experimented on the same corpus.

The INNTC method consist of two steps:

1. Clustering the training texts to some clusters.

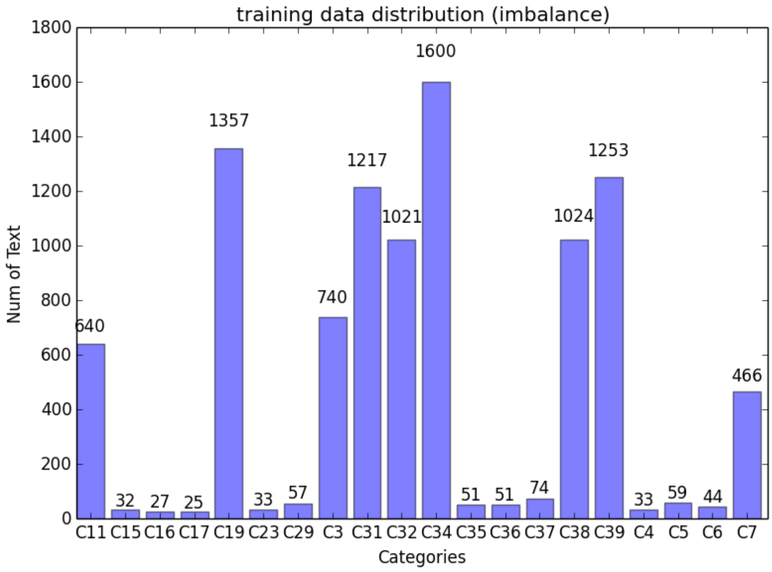


Figure 5.2: The distribution of training texts in Fudan Corpus [33].

2. Classifying the test texts with the clusters and KNN algorithm.

First, the training texts are converted to the vectors of numeric values by using TF-IDF algorithm. Then, the one-pass clustering algorithm to generate the clusters from the training texts. During the clustering process, a text is added to a clustering by using cosine similarity algorithm, if the text and the cluster have the same class label and the similarity is larger than the threshold. In other cases, the text is treated as a new cluster. When a new text is added into a cluster, this cluster have to update the weight of the features. The details about the steps are described as follows [18]:

(1) Initialize an empty set of clusters m_0, and read the first text p from the training texts.

(2) Generate a new cluster with the text p, and use the class label of the text p as the class label of the new cluster.

(3) If no texts are left in the training texts, go to step (6), otherwise read a new text p, compute the similarities between p and all the clusters $\bar{C}$ in m_0, and find the cluster C_i^0 in m_0 that is the closest to the text p.

(4) If $sim(p, C_i^0)) < threshold$ or the class labels are different between the text p and the nearest cluster, go to step (2).

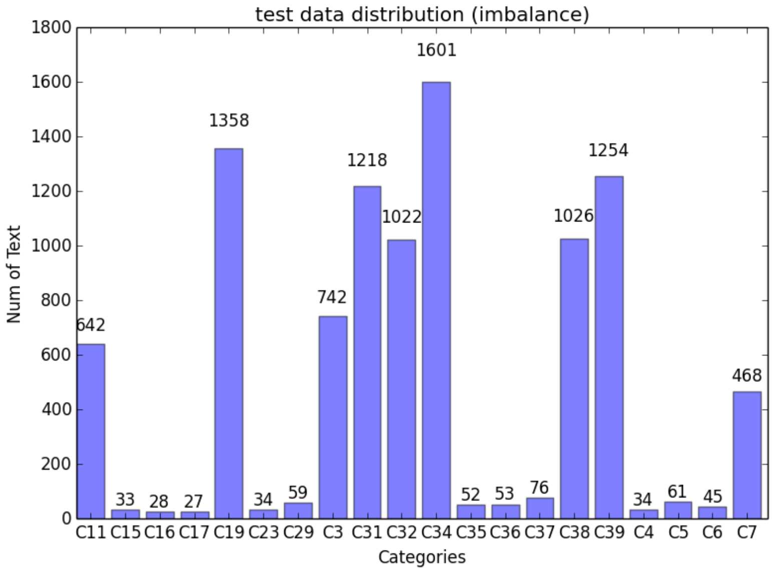


Figure 5.3: The distribution of test texts in Fudan Corpus [33].

(5) Combine text p into cluster (C_i^0) and update the weight of features for cluster C_i^0, go to step (3).

(6) Stop clustering, get the clustering results $m_0 = \{C_1^0, C_2^0, C_3^0, ..., C_n^0\}$, each cluster in m_0 is consisted of weighted features and its class label, and m_0 is the classification model for KNN method.

During the clustering process, each cluster is represented as a vector of numeric values which is the centroid vector for itself. The equation of updating the feature weights for the clusters in step (5) is described as follows [18]:

$$w_{c_i^0}^{j+1}(t) = \frac{w_{c_i^0}^{j}(t) \times |c_i^0| + w(t)_p}{|c_i^0| + 1}$$

where $w_{c_i^0}^{j+1}(t)$ indicates the new weight of feature t in cluster C_i^0; $w_{c_i^0}^{j}(t)$ is the weight of feature t in cluster C_i^0; $w(t)_p$ is the weight of feature t in text p; $|c_i^0|$ is the number of texts included in cluster C_i^0.

To classify the test texts, INNTC method used clusters as the training examples to classify testing examples by using KNN classifier. The details are described as follows: for a test text x, compute the score between each cluster in m_0 and the text x by using the following equation, and assign the class label of the cluster to the test text x, which has the highest score [18].

$$f(x) = \arg\max_{j} ClusterScore(x, C_j) = \sum_{C_i^0 \in kNN} sim(x, C_i^0) y(C_i^0, C_j)$$

where $f(x)$ is the class label assigned to the test text x; $ClusterScore(x, C_j)$ is the score between the candidate category C_j and the test text x; $sim(x, C_i^0)$ is the similarity between the test text x and the cluster C_i^0 in m_0; $y(C_i^0, C_j) \in 0, 1$ is the relationship between the cluster C_i^0 and the category C_j, ($y = 1$ indicates that cluster C_i^0 is in the category C_j, or $y = 0$).

In our experiments, INNTC method is used to classify the texts which are represented with the information from Baiku Baike.

5.1.3 Experiment Setting

The FUDAN corpus is used in the experiments as showed in Table 5.1, Figure 5.2 and Figure 5.3. The texts are split to the training texts and the test texts.

The Standford Chinese word segment tool is used to segment words in the texts. The words are used as the features. The features are weighted by TF-IDF algorithm. After weighting the features, the training texts are converted to the vectors of numeric values. Then, we enrich the text representation with the information form Baidu Baike which is used as the background knowledge. Finally, the training texts are represented as the vectors combined with TF-IDF values and the similarities between the texts and the background knowledge. Specially, to select the keywords for the categories, Standford POS tagging tool is used to tag the POS for the words.

The Linear SVM classifier is used to train the classification model with default parameter setting, which is provided by scikit-learn library.

In the predicting process, the test texts are preprocessed by the same steps on training texts, which are also converted to the vectors combined with TF-IDF values and the similarities between the texts and the background knowledge.

For the baseline method, the default setting is used, including the threshold, etc., which is elaborated in [18].

Because the category names are in English, but we need their Chinese names. To download the articles of category names from Baidu Baike, the category names are translated to Chinese. In some cases, one English word has more than one translations. Table 5.2 shows the Chinese category names translated from the English names.

Categories	Chinese names
C11-Space	太空 (space), 航天 (space flight)
C15-Energy	能源 (energy)
C16-Electronics	电子学 (electronics), 电子工业 (elecronic industry)
C17-Communication	电信业 (telecommunication industry), 通信业 (communication industry), 电信 (telecommunication), excluding the part of laws
C19-Computer	计算机 (computer), excluding the table of ages
C23-Mine	矿业 (mining industry), 矿产 (mineral)
C29-Transport	交通运输业 (transport industry)
C31-Enviornment	生态环境 (ecological environment)
C32-Agriculture	农业 (agriculture)
C34-Economy	经济 (economy)
C35-Law	法律 (law)
C36-Medical	医药行业 (medical industry), 医疗 (medical service)
C37-Military	军事 (military)
C38-Politics	政治 (politics)
C39-Sports	体育 (sports) excluding part of book
C3-Art	艺术 (art), including both concepts of 基本概念 (basic concept) and 文化名词 (proper noun related to culture)
C4-Literature	文学 (literature)
C5-Education	教育 (education)
C6-Philosophy	哲学 (philosophy)
C7-History	历史 (history)

Table 5.2: The Chinese category names of FUDAN corpus [33].

5.1.4 Experimental Results

In this section, the experimental results are presented, which have already presented in [33]. There are three group of results in this section, including the baseline method, the traditional method, and the proposed method.

The fist part presents the background knowledge, in which some keywords are selected from the training texts, and download their articles from Baidu Baike. The second part presents the comparison between the baseline and the proposed method.

The Background Knowledge from Baidu Baike

Table 5.3 shows the top 20 candidate words for each category in the training texts by ranking their CTF-ICF values. For each category, some words can not represent their own categories well. For an example, 参考 *refer* in the category *Space* and 标题 *title* in the category *Energy*, there is less semantic relation between these words and the categories. Therefore, some words which are less related to the categories should be filtered out. To filter out these words, the words and their category names are posted to Baidu Search Engine. By ranking the number of the search results, the top 3 words are selected, which are regarded as the keywords to present their categories.

Specially, when selecting the top 3 keywords, some words may have the same number of search results. In this case, the words with higher CTF-ICF values are selected.

Table 5.4 shows the category names which are used for selecting the top 3 keyword for each category. These category names and the candidate words are posted to Baidu Search Engine and ranked based on the number of the research results.

Specially, Table 5.4 is similar to Table 5.2. The difference between these two tables is that the category names are used for different purpose. in Table 5.2, some categories has more than one Chinese translation and they are used to download the articles of these category names from Baidu Baike. The articles within the same category are saved into the same text. Therefore, we get 20 texts for these categories as the background knowledge. In Table 5.4, these category names are used to select the top 3 words to represent their categories.

Table 5.5 presents the top 3 keywords for each category. The articles of these words are downloaded from Baidu Baike, and saved to texts with their class labels.

Totally, we get 79 articles with the category names and the top 3 keywords, include 59 keywords and 20 categories. The keyword 电子部 is not found in Baidu Baike.

The Results for Classifying Fudan Corpus

In the experiments, we evaluate the proposed method based on Baidu Baike with four groups of experiments.

These experiments are listed as following:

Categories	Top 20 candidate words ranked by CTF-ICF values
C11-Space	参考, 本文, 文献, 摘要, 主题词, 如图, 航空, 学报, 参数, 采用, 进行, 系数, 作者, 方法, 单位, 收稿, 分析, 误差, 计算, 给出
C15-Energy	标题, 日期, 作者, 电力, 记者, 开发, 国家, 发电, 发展, 技术, 温室, 电厂, 环境, 资源, 进行, 我国, 环保, 燃料, 全球, 利用
C16-Electronics	标题, 日期, 作者, 电子, 芯片, 技术, 电路, 公司, 发展, 市场, 记者, 国家, 显像管, 半导体, 企业, 世界, 成为, 工业, 电子部, 生产
C17-Communication	通信, 标题, 作者, 邮电部, 日期, 通信网, 邮电, 移动, 电话, 发展, 用户, 公司, 我国, 专网, 建设, 市话, 光缆, SDH, 容量, 业务
C19-Computer	本文, 参考, 文献, 学报, 定义, 给出, 函数, 算法, 摘要, 如图, 描述, 模型, 方法, 参数, 系统, 应用, 引言, 软件, 对应, 数据
C23-Mine	储量, 标题, 日期, 矿山, 作者, 生产, 采矿, 国家, 矿区, 金矿, 开采, 矿种, 资源, 矿产, 记者, 公司, 发展, 开发, 企业, 政策
C29-Transport	铁路, 交通部, 运输, 交通, 记者, 通车, 标题, 日期, 干线, 作者, 发展, 工程, 开行, 客运, 公路, 铁道部, 列车, 铁路局, 通讯员, 建成
C31-Enviornment	文献, 参考, 摘要, 浓度, 环境, 污染物, 研究, 本文, 进行, 科学, 简介, 学报, 方法, 作者, 出版社, 分析, 单位, 收稿, 影响, 采用
C32-Agriculture	农产品, 期号, 出处, 复印, 分类号, 原文, 农民, 地名, 耕地, 农村, 农户, 生产, 作者, 页号, 作物, 发展, 提高, 粮食, 增产, 进行
C34-Economy	期号, 出处, 复印, 分类号, 原文, 地名, 简介, 资本, 作者, 正文, 标题, 市场, 发展, 企业, 资产, 生产, 国家, 产业, 货币, 问题
C35-Law	发布, 合同法, 人民, 国家, 民事, 规定, 定本, 共和国, 合同, 国务院, 部门, 企业, 当事人, 必须, 机关, 订阅, 自治区, 管理, 应当, 直辖市
C36-Medical	治疗, 患者, 医院, 疗效, 临床, 记者, 研究, 病人, 疾病, 进行, 卫生, 专家, 外科, 卫生部, 手术, 组织, 医科, 单位, 医生, 医学
C37-Military	军队, 武装, 部队, 进行, 裁军, 问题, 记者, 武器, 举行, 打死, 导弹, 军备, 总统, 报道, 消减, 开始, 表示, 部署, 战斗机, 战斗
C38-Politics	期号, 出处, 复印, 分类号, 原文, 地名, 页号, 问题, 正文, 国家, 发展, 权力, 西方, 进行, 标题, 主义, 人民, 思想, 作者, 关系
C39-Sports	期号, 出处, 复印, 原文, 分类号, 页号, 地名, 比赛, 进行, 运动员, 作者, 发展, 运动, 方法, 正文, 实践, 提高, 研究, 训练, 方面
C3-Art	期号, 出处, 复印, 分类号, 原文, 创作, 作品, 地名, 文艺, 人物, 作者, 表现, 标题, 时代, 艺术家, 小说, 思想, 作家, 页号, 正文
C4-Literature	文化, 作者, 标题, 日期, 民族, 出版, 传统, 出版社, 文明, 文艺, 表现, 文物, 精神, 影响, 发展, 方面, 作品, 西方, 观众, 社会
C5-Education	学校, 学生, 教师, 工作, 培养, 记者, 国家, 思想, 学习, 发展, 中小学, 社会, 作者, 数学, 提高, 办学, 标题, 日期, 家长, 教委
C6-Philosophy	思想, 理论, 主义, 标题, 问题, 实践, 社会主义, 认识, 进行, 社会, 作者, 规律, 发展, 唯物, 日期, 辩证法, 精神, 科学, 基础, 发信人
C7-History	期号, 出处, 复印, 分类号, 原文, 地名, 页号, 正文, 作者, 人物, 标题, 时代, 统治, 注释, 思想, 表现, 发展, 社会, 文化, 阶级

Table 5.3: Top 20 candidate words of each category ranked by the CTF-ICF values [33].

(1) The Linear SVM method

Categories	category names
C11-Space	航天 (space flight)
C15-Energy	能源 (energy)
C16-Electronics	电子工业 (elecronic industry)
C17-Communication	通信业 (communication industry)
C19-Computer	计算机 (computer)
C23-Mine	矿业 (mining industry)
C29-Transport	交通运输业 (transport industry)
C31-Enviornment	生态环境 (ecological environment)
C32-Agriculture	农业 (agriculture)
C34-Economy	经济 (economy)
C35-Law	法律 (law)
C36-Medical	医疗 (medical service)
C37-Military	军事 (military)
C38-Politics	政治 (politics)
C39-Sports	体育 (sports)
C3-Art	艺术 (art)
C4-Literature	文学 (literature)
C5-Education	教育 (education)
C6-Philosophy	哲学 (philosophy)
C7-History	历史 (history)

Table 5.4: The category names for selecting top 3 keywords [33].

(2) Category_names + The Linear SVM method

(3) Category_names + Top_3_keywords + method (baseline)

(4) Category_names + Top_3_keywords + The Linear SVM method

In this list, the *Category_names* indicates computing the similarities between the texts and the articles of the category names from Baidu Baike, and enrich the text representation with these similarities. The *Top_3_keywords* indicates computing the similarities between the texts and the articles of the top 3 keywords of each category from Baidu Baike, and enrich the text representation with these similarities.

At the beginning, we only use the articles of the *category names* to enrich the text representation. The results showed that the macro-F1 score is improved, but the precision decreased a little with the Linear SVM classifier. To improve both macro F1 score and precision, the articles of the top 3 keywords of each category are used to enrich the text representation.

Table 5.6 presents the result comparison between the baseline and the proposed method. The *+baidu* means enriching the text representation with Baidu Baike. Table 5.7 shows

Categories	Top 3 keywords
C11-Space	航空 (aviation), 系数 (coefficient), 进行 (perform)
C15-Energy	电力 (power), 电厂 (power plant), 资源 (resource)
C16-Electronics	电子(elecron), 工业(industry), 电子部 (ministry of elecronic industry)
C17-Communication	移动 (mobile), 通信 (communication), 通信网 (communications network)
C19-Computer	定义 (definition), 算法 (algorithm), 描述 (describe)
C23-Mine	矿山 (mine), 采矿 (mining), 金矿 (gold ore)
C29-Transport	铁路 (railway), 公路 (highway), 交通 (traffic)
C31-Enviornment	本文 (this article), 环境 (environment), 科学 (science)
C32-Agriculture	农村 (village), 生产 (produce), 农民 (farmer)
C34-Economy	资本 (capital), 发展 (develop), 国家 (nation)
C35-Law	发布 (publish), 规定 (rule), 民事 (civil)
C36-Medical	治疗 (treatment), 临床 (clinical), 病人 (patient)
C37-Military	军队 (army), 部队 (troop), 武器 (weapon)
C38-Politics	发展 (develop), 权力 (authority), 思想 (thought)
C39-Sports	比赛 (match), 运动员 (athlete), 运动 (sports)
C3-Art	创作 (creation), 艺术家 (artist), 小说 (fiction)
C4-Literature	出版 (publish), 传统 (tradition), 文明 (civilization)
C5-Education	学校 (school), 学生 (student), 教师 (teacher)
C6-Philosophy	认识 (realization), 科学 (science), 思想 (thought)
C7-History	文化 (culture), 时代 (age), 思想 (thought)

Table 5.5: The top 3 keywords for each category. [33]

the impact of enriching the text representation with Baidu Baike. The *+label* means using the articles of the category names to compute the similarities. Figure 5.4 presens the overview of experimental results with marco precision, recall and F1 score.

5.1.5 Discussion

The experimental results show that the proposed method outperforms over the traditional methods. Further, the results indicate that the background knowledge from Baidu Baike bring the positive impact to the experiments. This method could be regarded as an alternative method for text classification task.

	Precision			Recall			F1 score		
	inntc +baidu (baseline)	linear SVM	linear SVM +baidu	inntc +baidu (baseline)	linear SVM	linear SVM +baidu	inntc +baidu (baseline)	linear SVM	linear SVM +baidu
C11	0.93	0.95	**0.95**	0.81	0.93	**0.93**	0.87	0.94	**0.94**
C15	1.00	1.00	**1.00**	0.42	0.55	**0.61**	0.60	0.71	**0.75**
C16	1.00	1.00	**1.00**	0.04	0.54	**0.54**	0.07	0.70	**0.70**
C17	0.76	**0.87**	0.84	0.59	0.74	**0.78**	0.67	0.80	**0.81**
C19	0.93	0.96	**0.96**	0.96	0.99	**0.99**	0.94	0.98	**0.98**
C23	0.87	0.89	**0.93**	0.38	0.71	**0.74**	0.53	0.79	**0.82**
C29	**0.95**	0.88	0.88	0.61	0.88	**0.90**	0.74	0.88	**0.89**
C3	0.86	0.89	**0.89**	0.89	0.95	**0.95**	0.87	0.92	**0.92**
C31	0.94	0.96	**0.96**	0.90	0.97	**0.97**	0.92	0.96	**0.97**
C32	0.88	0.95	**0.95**	0.94	0.96	**0.96**	0.91	0.95	**0.95**
C34	0.81	0.93	**0.93**	0.94	0.96	**0.96**	0.87	0.94	**0.95**
C35	**1.00**	0.80	0.82	0.37	0.63	**0.63**	0.54	0.71	**0.72**
C36	1.00	1.00	**1.00**	0.36	0.74	**0.77**	0.53	0.85	**0.87**
C37	**0.96**	0.90	0.88	0.33	0.75	**0.76**	0.49	0.82	**0.82**
C38	0.83	0.91	**0.92**	0.92	0.93	**0.93**	0.87	0.92	**0.93**
C39	0.91	0.95	**0.95**	0.95	0.98	**0.98**	0.93	0.96	**0.97**
C4	0.00	0.71	**0.71**	0.00	0.15	**0.15**	0.00	0.24	**0.24**
C5	0.50	0.75	**0.75**	0.02	0.25	**0.30**	0.03	0.37	**0.42**
C6	0.79	0.86	**0.88**	0.24	0.42	**0.47**	0.37	0.57	**0.61**
C7	0.76	0.85	**0.85**	0.59	0.78	**0.79**	0.66	0.81	**0.82**

Table 5.6: The comparison of results with the baseline [33].

Specially, Fudan corpus is an imbalanced dataset. The biggest category contains more than 1,000 texts, but the smallest category contains only tens of texts. After representing the texts with TF-IDF values, the length of the vector is more than 350,000 words.

In Table 5.3, some of the top 20 words do not have the relation with their categories by reading the texts in each category.. However, these words have the high CTF-ICF values in FUDAN corpus. Therefore, those unrelated words in Table 5.3 have to be filtered out. By selecting the top 3 keywords for each category, the words are the most related to their categories in Table 5.5. As mentioned above, the keyword 电子部 “ministry of electronic industry”is not found in Baidu Baike. Therefore, there are 79 concepts in the experiment (59 articles from the keywords, 20 articles from the category names).

Figure 5.4 shows that the proposed approach improve the effectiveness for text classification. We compare the macro results with the different methods. The first group of results is from the proposed method based on Baidu Baike. The second group of results is

	Precision			Recall			F1 score		
	linear SVM	linear SVM +lable	linear SVM +lable +top3 words	linear SVM	linear SVM +lable	linear SVM +lable +top3 words	linear SVM	linear SVM +lable	linear SVM +lable +top3 words
C11	0.95	0.94	**0.95**	0.93	0.92	**0.93**	0.94	0.93	**0.94**
C15	1.00	1.00	**1.00**	0.55	0.61	**0.61**	0.71	0.75	**0.75**
C16	1.00	0.94	**1.00**	0.54	0.54	**0.54**	0.70	0.68	**0.70**
C17	**0.87**	0.84	0.84	0.74	0.78	**0.78**	0.80	0.81	**0.81**
C19	0.96	0.96	**0.96**	0.99	0.99	**0.99**	0.98	0.97	**0.98**
C23	0.89	0.93	**0.93**	0.71	**0.76**	0.74	0.79	**0.84**	0.82
C29	0.88	0.87	**0.88**	0.88	0.88	**0.90**	0.88	0.87	**0.89**
C3	0.89	0.89	**0.89**	0.95	0.95	**0.95**	0.92	0.92	**0.92**
C31	0.96	0.96	**0.96**	0.97	0.97	**0.97**	0.96	0.97	**0.97**
C32	0.95	0.94	**0.95**	0.96	0.96	**0.96**	0.95	0.95	**0.95**
C34	0.93	0.93	**0.93**	0.96	0.96	**0.96**	0.94	0.95	**0.95**
C35	0.80	0.80	**0.82**	0.63	0.63	**0.63**	0.71	0.71	**0.72**
C36	1.00	1.00	**1.00**	0.74	0.77	**0.77**	0.85	0.87	**0.87**
C37	**0.90**	0.89	0.88	0.75	0.75	**0.76**	0.82	0.81	**0.82**
C38	0.91	0.92	**0.92**	0.93	0.93	**0.93**	0.92	0.92	**0.93**
C39	0.95	0.95	**0.95**	0.98	0.98	**0.98**	0.96	0.97	**0.97**
C4	0.71	0.71	**0.71**	0.15	0.15	**0.15**	0.24	0.24	**0.24**
C5	0.75	0.70	**0.75**	0.25	0.26	**0.30**	0.37	0.38	**0.42**
C6	0.86	0.83	**0.88**	0.42	0.44	**0.47**	0.57	0.58	**0.61**
C7	0.85	0.85	**0.85**	0.78	0.79	**0.79**	0.81	0.82	**0.82**

Table 5.7: The comparison of results based on the top 3 keywords [33].

from the traditional method without the background knowledge. The last group of results is from the baseline method. The results show that the proposed method outperforms over the traditional methods by using Linear SVM classifier, with macro precision (90.31%), recall (75.45%), F1 score (80.32%), which are improved with 0.02%, 0.15%, 0.12%. The recall and F_1 score are improved obviously. This indicates that the background knowledge can complement the common information between the training and test texts.

In the traditional method, the texts are represented as the vectors that has around 350,000 features, each feature represents a word. In the proposed method, 79 similarities are added to each vector. The improvement is benefited from the 79 similarities, although 79 similarities are very small, comparing to 350,000 features. This indicates these similarities are the efficient features for representing texts.

Note that, the number of search results is not always the same from Baidu Search

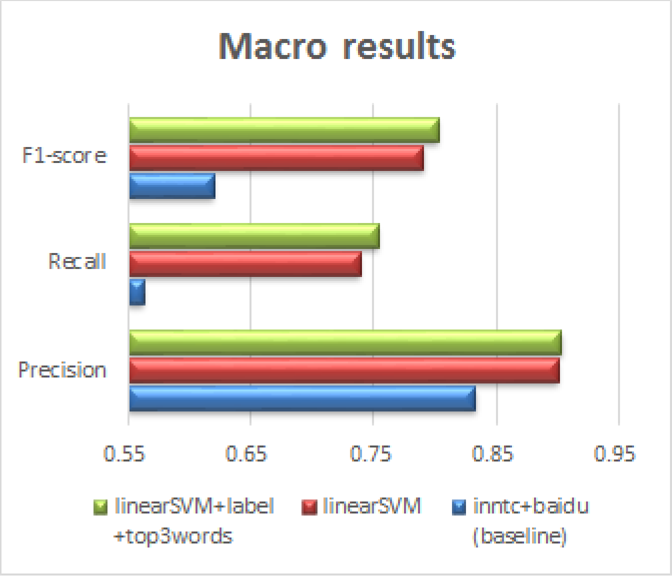


Figure 5.4: The comparison of macro precision, recall, F1 score [33].

Engine, and the maximum number of results is always 100,000,000. Therefore, the top 3 keywords of each category sometimes may change.

Additionally, to investigate the robust of the proposed method, some further experiments are performed with different proportion of the training and test texts. The different proportions of training/testing texts include 2:8, 3:7, 3.3:6.7, 5:5, 6.7:3.3, 7:3, and 8:2. For each kind of these proportion, we perform 10 experiments. The training and test texts are selected randomly in each time. We compare the results between using TF-IDF values only and the proposed method.

Figure 5.5, 5.6 and 5.7 show the results of macro F_1 score, macro precision, and macro recall respectively. These results confirm that the proposed method could obviously improve F_1 score and recall with imbalanced data for text classification task, while keeping precision stable. When the proportion is 8:2, all of F_1 score, precision, and recall get improvement. Moreover, the improvement of F_1 score and recall is rather obvious when the training set is much smaller than the testing set. This shows that the proposed method performance better when there are only a little training texts. This also indicates that using the background knowledge could complement the information between the training and test texts.

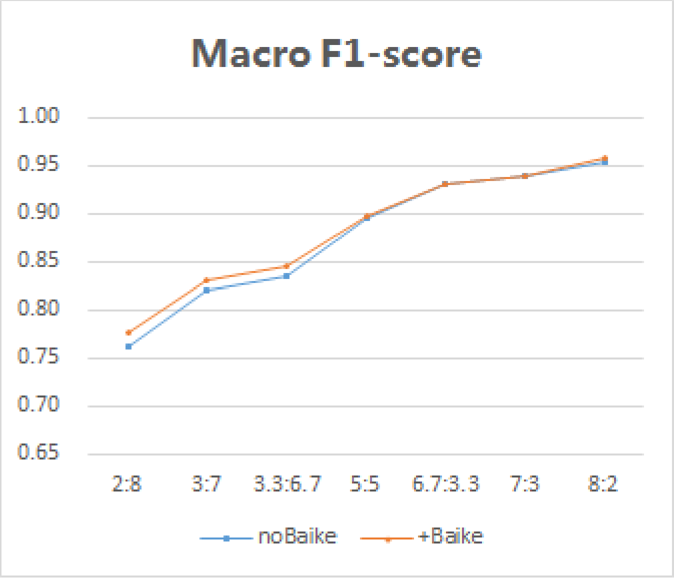


Figure 5.5: The macro F1 with different proportions of training/testing texts.

5.2 Classifying texts with Character Co-occurrence

The performance of the method based on character co-occurrence is evaluated with two Chinese text classification corpora, FUDAN corpus and Sougou classification corpus in the reduced version. Two additional corpora are used as the background knowledge, People's daily news and SougouCA news.

Section 5.2.1 introduces the datasets used in the experiments. Section 5.2.2 presents the experimental setting. Finally, the experimental results are showed and discussed in section 5.2.3 and section 5.2.4.

5.2.1 Dataset

FUDAN corpus is used to train classification model and predict a given text. This corpus have been introduce in the section 5.1.1

Table 5.1 shows the categories of FUDAN corpus. Figure 5.1 show the the distribution of the texts in each categories in FUDAN corpus. The x-axis indicates the categories. The y-axis indicates the number of texts. At the top of each bar, it is the number of texts in the category.

In the experiments, we also used another corpus. Sougou classification corpus in the

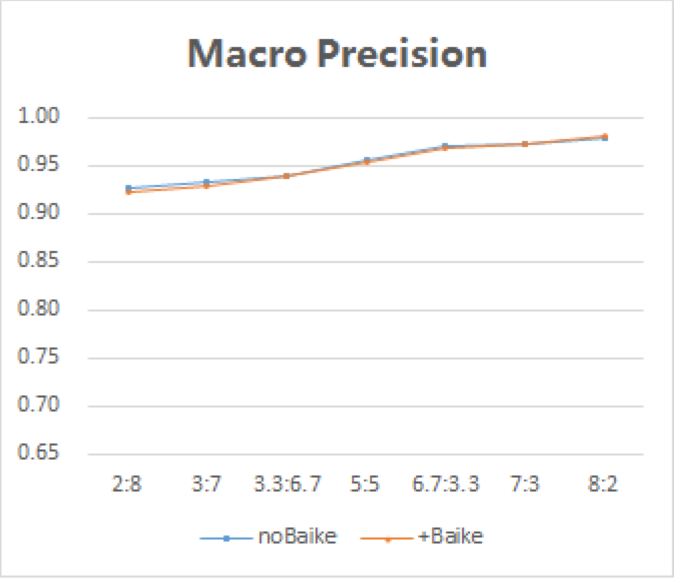


Figure 5.6: The macro precision with different proportions of training/testing texts.

reduce version[3] is a balanced text classification corpus, containing 17910 within 9 categories, in which each category has the same number of texts. In this section, we will use the *Sougou corpus* for short. Table 5.8 shows the categories of Sougou corpus. Figure 5 shows the distribution of the texts in Sougou corpus.

C08-Economy	C10-IT	C13-Health
C14-Sports	C16-Tour	C20-Education
C22-Recruitment	C23-Culture	C24-Military

Table 5.8: The categories in Sougou corpus.

To use character co-occurrence as the background knowledge, two additional corpus are used in the experiments. They are People's daily news and SougouCA news. The purpose of using these two corpus is to investigate whether the difference background knowledge can impact the results.

People's daily news[4] is downloaded from Internet, which is used for computing the character co-occurrence, and then used as the background knowledge. After the preprocessing, the People's daily news contains 61,213,647 sentences, the average length of the sentences is 8.64 characters, and the size of the file is around 1.6G bits.

The texts from SougouCA news[5] [48] is used for computing the second character

[3] http://www.sogou.com/labs/dl/c.html
[4] http://paper.people.com.cn
[5] http://www.sogou.com/labs/resource/ca.php

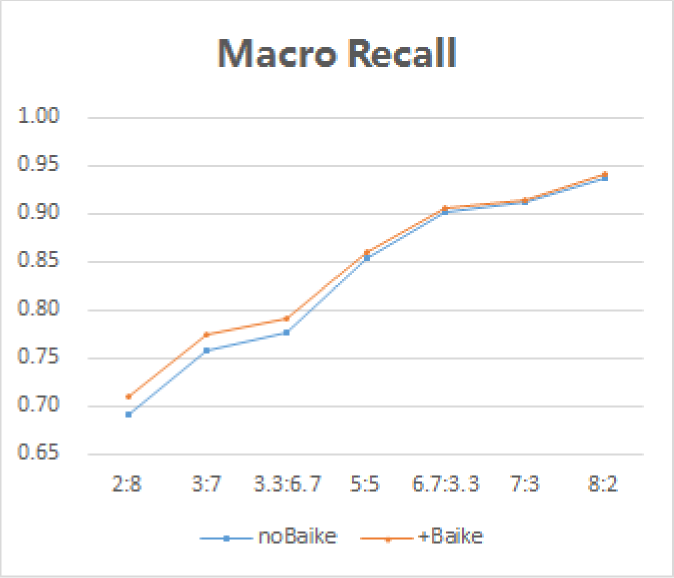


Figure 5.7: The macro recall with different proportions of training/testing texts.

co-occurrence, and then used as the background knowledge. After the preprocessing, SougouCA news contains 68,034,919 sentences, the average length of sentences is 8.06 characters, and the size of the file is around 1.6G bits.

5.2.2 Experiment Setting

First of all, the additional corpora are used to compute two character co-occurrence matrices separately according to the steps described in Section 4.3. The character co-occurrence matrices are regarded as the background knowledge. To select the top N key characters, the training texts in FUDAN corpus and Sougou corpus are segmented by Stanford word Segment tool, and tagged by Stanford POS tagger. Then, each text is segmented by the characters. The texts in the same category are combined together. For an example, there are 3 categories in the corpus, after combining, there are 3 texts for each category. The TF-IDF method is used to weight the characters for each category. The top N key characters of each category are selected by ranking their weight. The character co-occurrence of the top N key characters are used as the background knowledge. In the experiments, the top N is set from 0 to 20, $N \in [0, 20]$ ($N = 0$ means not using the background knowledge. $N > 0$ means using the specific number of key characters to add the background knowledge).

The character co-occurrence is also computed for the training and test texts. To get the

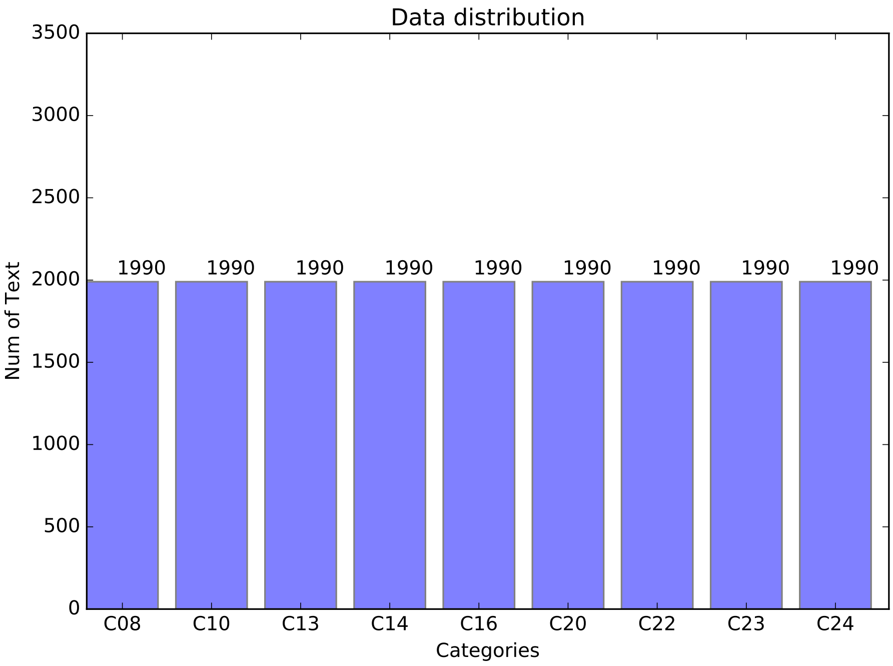


Figure 5.8: The distribution of texts in Sougou corpus.

information from the background knowledge, the similarities of character co-occurrence between the texts and the background knowledge are computed. Specially, the character co-occurrence in the different distances is also involved in our experiments, the length of the distance is from 1 to 6. Thus, six character co-occurrence matrices would be computed for both the texts and the additional corpora with the different distances between two characters.

For training classification model and prediction, we combine the text representation with TF-IDF values and the similarities of character co-occurrence as showed in Figure 4.5. And the Linear SVM classifier that provided by scikit-learn library is used to train the classifier with default parameter setting in 5-folds cross validation. The traditional method without character co-occurrence information is also conducted in 5-folds cross validation by the Linear SVM classifier with default parameter setting for comparing the experimental results.

To evaluate performance, the precision, recall, and F1 measure mentioned above, are used. In the experiment, the macro precision, recall, and F1 measure are computed by the functions provided by scikit-learn library.

5.2.3 Experimental Results

In this section, the experimental results are presented. The first part shows the top 20 key characters for each category in the two classification corpora. The second part presents the results of the proposed method with FUDAN corpus. The third part presents the results of the proposed method with Sougou corpus.

Top 20 key characters for each category

Table 5.9 and 5.10 show the top 20 key characters for each category in FUDAN corpus and Sougou corpus separately. In these two tables, each row represents a category, and the characters are listed from the top 1 to 20, ranking by their weight. Specially, some characters repeat in the different categories. When computing the character co-occurrence with these key characters, each character only appears once in the set of key characters.

Additionally, after obtaining the character co-occurrence matrices from these two external corpora, 8887 distinct characters are extracted from the People's daily news, 6526 distinct characters are extracted from the SougouCA news. Therefore, with the different additional corpora, the dimensions of these two character co-occurrence matrices are different. After removing the repetitive characters, totally there are 142 key characters in Fudan corpus and 99 key characters in Sougou corpus.

The Results for Classifying Fudan Corpus

To evaluate the proposed method with Fudan Corpus, three groups of experiments are conducted, which are listed as follows:

(1) FUDAN classification corpus + People's daily news

(2) FUDAN classification corpus + SougouCA news

(3) FUDAN classification corpus + People's daily news + SougouCA news

In FUDAN corpus, each text has its class label. The People's daily news and SougouCA news corpus are the additional corpora, which are used for building the background knowledge of character co-occurrence. Each group of experiments is conducted in 5-fold cross

Categories	Top 20 key characters
C11-Space	度, 数, 量, 系, 机, 方, 动, 性, 工, 力, 面, 图, 统, 程, 件, 结, 测, 流, 计, 法
C15-Energy	电, 能, 源, 国, 发, 人, 力, 工, 家, 油, 核, 作, 者, 会, 气, 业, 技, 量, 机, 生
C16-Electronics	电, 子, 产, 国, 业, 机, 技, 公, 司, 术, 工, 品, 者, 人, 信, 发, 企, 生, 家, 场
C17-Communication	信, 通, 电, 国, 业, 网, 公, 司, 邮, 产, 经, 人, 技, 话, 设, 者, 机, 市, 发, 务
C19-Computer	数, 系, 统, 法, 方, 程, 算, 性, 理, 模, 件, 图, 器, 据, 文, 结, 用, 机, 网, 信
C23-Mine	矿, 国, 产, 地, 工, 业, 金, 资, 煤, 源, 量, 人, 石, 发, 家, 作, 区, 部, 经, 者
C29-Transport	路, 车, 运, 铁, 工, 国, 通, 交, 公, 部, 客, 输, 道, 人, 航, 线, 程, 业, 局, 地
C3-Art	文, 艺, 人, 学, 术, 作, 性, 理, 论, 生, 化, 主, 义, 方, 诗, 者, 体, 家, 形, 史
C32-Agriculture	农, 业, 产, 经, 国, 生, 地, 化, 济, 品, 资, 发, 市, 工, 场, 技, 民, 力, 展, 政
C34-Economy	经, 济, 业, 国, 产, 资, 政, 市, 制, 会, 人, 企, 场, 社, 发, 力, 展, 家, 主, 工
C35-Law	国, 人, 业, 法, 合, 同, 工, 品, 产, 理, 管, 经, 部, 家, 企, 地, 技, 计, 术, 民
C36-Medical	医, 人, 生, 病, 药, 疗, 国, 学, 会, 家, 院, 者, 科, 电, 卫, 术, 员, 癌, 中, 工
C37-Military	军, 机, 战, 国, 人, 部, 队, 地, 空, 力, 弹, 海, 武, 事, 兵, 会, 方, 装, 区, 员
C38-Politics	政, 治, 主, 国, 会, 人, 社, 民, 义, 制, 经, 家, 济, 学, 党, 方, 理, 体, 力, 权
C39-Sports	学, 育, 教, 体, 人, 动, 生, 国, 会, 理, 作, 文, 社, 方, 运, 力, 校, 科, 业, 性
C4-Literature	文, 化, 学, 人, 国, 史, 民, 会, 作, 方, 者, 社, 族, 家, 中, 历, 术, 书, 地, 主
C5-Education	教, 学, 育, 生, 人, 校, 国, 会, 业, 家, 工, 社, 中, 作, 子, 师, 孩, 方, 地, 义
C6-Philosophy	学, 哲, 人, 主, 会, 义, 社, 理, 文, 思, 论, 方, 题, 作, 者, 想, 民, 国, 生,工
C7-History	史, 人, 学, 文, 历, 主, 国, 民, 义, 会, 作, 地, 方, 社, 化, 政, 理, 家, 者, 论
C31-Environment	水, 物, 环, 量, 度, 境, 化, 学, 生, 地, 土, 工, 性, 污, 理, 浓, 程, 分, 体, 方

Table 5.9: Top 20 key characters of each category in FUDAN corpus

Categories	Top 20 key characters
C08- Economy	股, 公, 司, 资, 会, 人, 市, 业, 行, 东, 权, 金, 产, 场, 议, 投, 事, 价, 方, 证
C10-IT	业, 电, 网, 人, 公, 司, 机, 务, 市, 产, 国, 用, 信, 场, 品, 商, 户, 者, 方, 家
C13-Health	人, 医, 药, 生, 病, 品, 性, 者, 体, 业, 院, 物, 家, 心, 疗, 国, 中, 子, 方, 女
C14-Sports	赛, 球, 队, 场, 员, 比, 人, 国, 播, 主, 直, 手, 时, 分, 体, 力, 联, 军, 中, 足
C16-Tour	游, 旅, 人, 地, 国, 客, 市, 行, 区, 场, 家, 公, 业, 者, 方, 民, 航, 机, 城, 中
C20-Education	学, 生, 考, 人, 业, 校, 教, 题, 大, 专, 理, 试, 科, 育, 高, 国, 法, 工, 子, 文
C22-Recruitment	人, 业, 工, 生, 公, 作, 学, 职, 司, 理, 企, 员, 力, 事, 时, 者, 会, 大, 位, 面
C23-Culture	人, 国, 文, 学, 家, 子, 民, 生, 地, 会, 大, 中, 者, 主, 方, 时, 军, 事, 化, 作
C24-Military	军, 战, 机, 国, 部, 队, 空, 人, 力, 事, 海, 弹, 作, 地, 方, 导, 防, 演, 兵, 系

Table 5.10: Top 20 key characters of each category in Sougou classificaion corpus

validation with the different distances between characters, and using several different distances together. Therefore, around 1,100 ($5 \times (6 + 5) \times 21$) experiments are conducted in each group of experiments. When using the traditional method, the results are around 0.901 macro precision, 0.760 macro recall, 0.804 macro F1 score on FUDAN corpus. Each group of results is calculated by *cross_validation.cross_val_score()* function which is provided by scikit-learn library, with the parameter setting, *scoring = 'precision_macro' or 'recall_macro'*, and *cv = 5.*

Figure 5.9 compares the results with FUDAN corpus and People's daily news. Each curve represents a group of the experiments. In each group, the experiments are conducted with the character co-occurrence of the specific distance between characters and the different number of key characters. To enrich text representation, the similarities of character co-occurrence between the texts and the background knowledge are added to the feature vectors of the texts. The legend in Figure 5.9 represents the groups of distances used in the experiments. For an example, D1-1 means that two characters are adjacent; D2-2 means that two characters are separated by a character. For other example, D1-3 means that three kinds of distances between characters are used in the experiments; Three groups of similarities of character co-occurrence are computed referring to the background knowledge for each text; Finally, all these groups of similarities are added to the feature vectors of each text. Other details are introduced as following:

- D1-1: Combining the similarities of the character co-occurrence with $distance = 1$, for an example, “大” and “学 ” in “大学”.
- D2-2: The distance is 2 between characters, for an example, “研” and “室 ” in “研究室”.
- D3-3: The distance is 3 between characters.
- D4-4: The distance is 4 between characters.
- D5-5: The distance is 5 between characters.
- D6-6: The distance is 6 between characters.
- D1-2: Combining the similarities of the character co-occurrence with $distance = 1$ and $distance = 2$
- D1-3: Combining the similarities of the character co-occurrence with $distance \in [1, 3]$
- D1-4: Combining the similarities of the character co-occurrence with $distance \in [1, 4]$
- D1-5: Combining the similarities of the character co-occurrence with $distance \in [1, 5]$
- D1-6: Combining the similarities of the character co-occurrence with $distance \in [1, 6]$

The y axis indicates the values of precision, recall and F1 score. The x axis represents how many key characters of each category are used as the features of character co-occurrence. Specially, when N is set to 0, it means that only the TF-IDF values are used to train classifier. At the beginning of each curve, N is 0.

In the Figure 5.9, macro recall and F1 score could be improved the by the proposed approach obviously. Besides, precision also could be improved by setting the distance to 2, 3, 4 and 5, with top 1 key character. Specially, when setting the distance to 6, the best macro precision is 0.911 (+1%) with top 2 key characters. The results indicate that adding the character co-occurrence information could improve the effectiveness in text classification task.

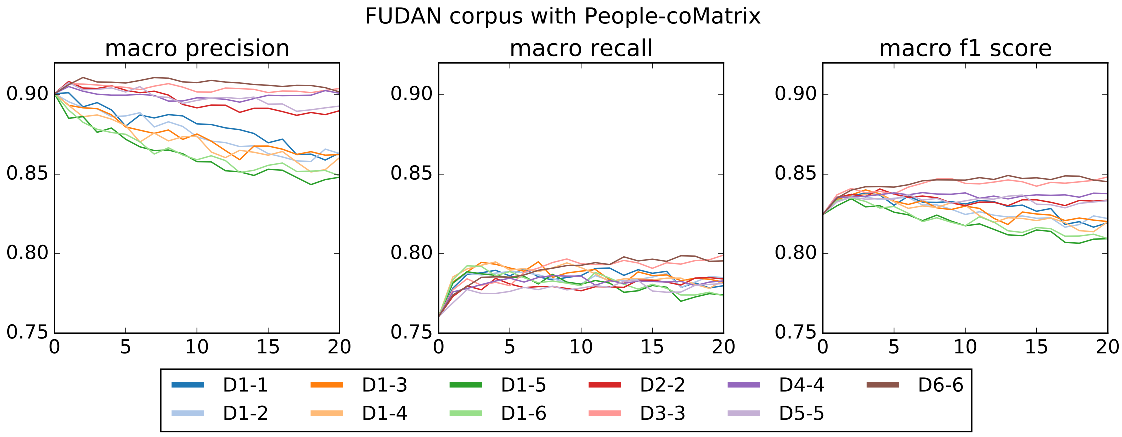


Figure 5.9: Comparison on FUDAN corpus with character co-occurrence similarity based on People's daily news.

Figure 5.10 compares the results with FUDAN corpus and People's daily news. Figure 5.11 shows the performance comparison with FUDAN corpus and character co-occurrence similarities referred to People's daily news and SougouCA news. In these two groups of experiments, the macro recall and F1 score are improved obviously. By setting the distance to 2, 3, 4, 5, and 6, with top 1 or 2 key characters, the precision is also be improved.

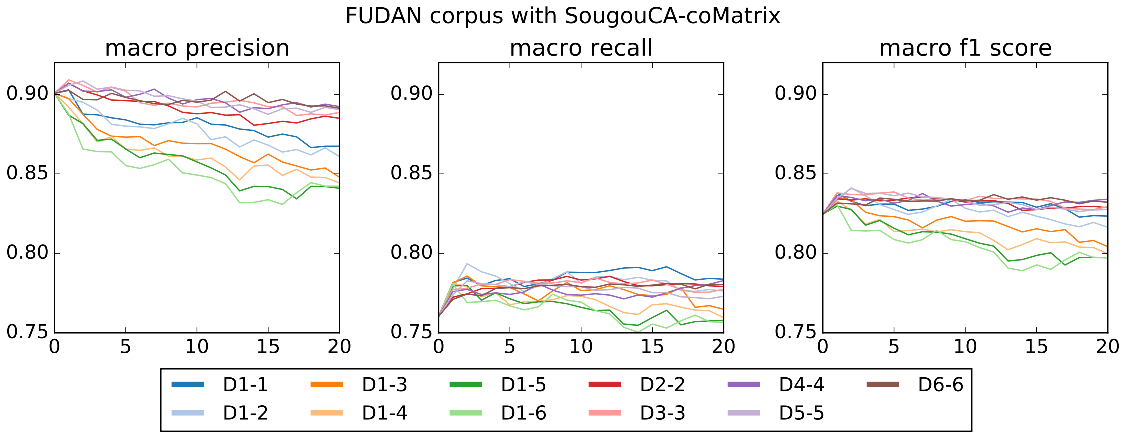


Figure 5.10: Comparison on FUDAN corpus with character co-occurrence similarity based on SougouCA news.

The Results for Classifying Sougou Corpus

To evaluate the proposed method with Sougou Corpus, three groups of experiments are conducted, which are listed as follows:

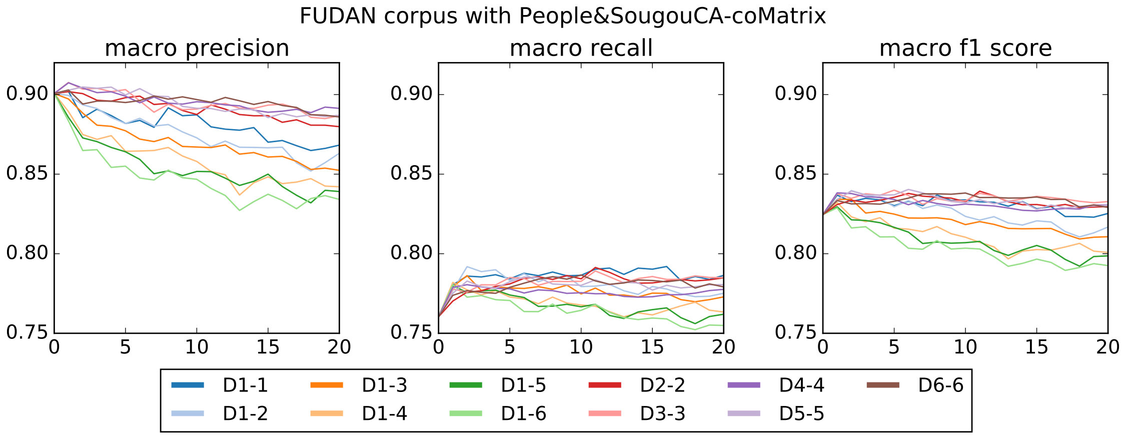


Figure 5.11: Comparison on FUDAN corpus with character co-occurrence similarity based on People's daily news and SougouCA news.

(1) Sougou classification corpus + People's daily news

(2) Sougou classification corpus + SougouCA news

(3) Sougou classification corpus + People's daily news + SougouCA news

In Sougou corpus, each text has its class label. The People's daily news and SougouCA news are the additional corpora, which are used for building the background knowledge of character co-occurrence. Each group of experiments is conducted in 5-fold cross validation with the different distances between characters from 1 to 6, and using several different distances together. Therefore, around 1,100 ($5 \times (6+5) \times 21$) experiments are conducted in each group of experiments. When using the traditional method, the results are around 0.893 macro precision, 0.892 macro Recall, 0.892 macro F1 score. Each group of results is calculated by *cross_validation.cross_val_score()* function which is provided by scikit-learn library, with the parameter setting, *scoring = 'precision_macro' or 'recall_macro'*, and *cv = 5.*

Figure 5.12 compares the results with Sougou classification corpus and character co-occurrence similarities based on People's daily news. In this group of experiments, the macro precision, recall, F1 score are improved when the distance between characters is set to 3.

Figure 5.13 compares the results with Sougou corpus and SougouCA news. Figure5.14 compares the results with Sougou corpus, People's daily news and SougouCA news. The

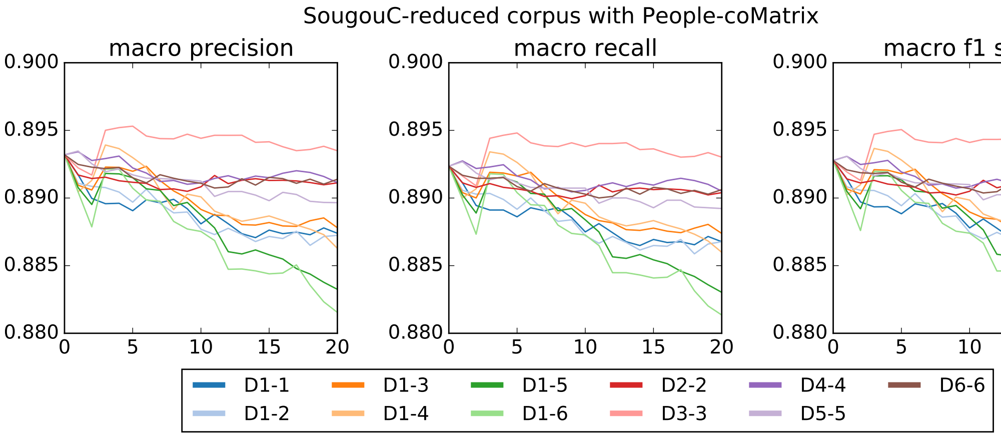


Figure 5.12: Comparison on Sougou classification corpus with character co-occurrence similarity based on People's daily news.

performance could not be improved by any parameter setting.

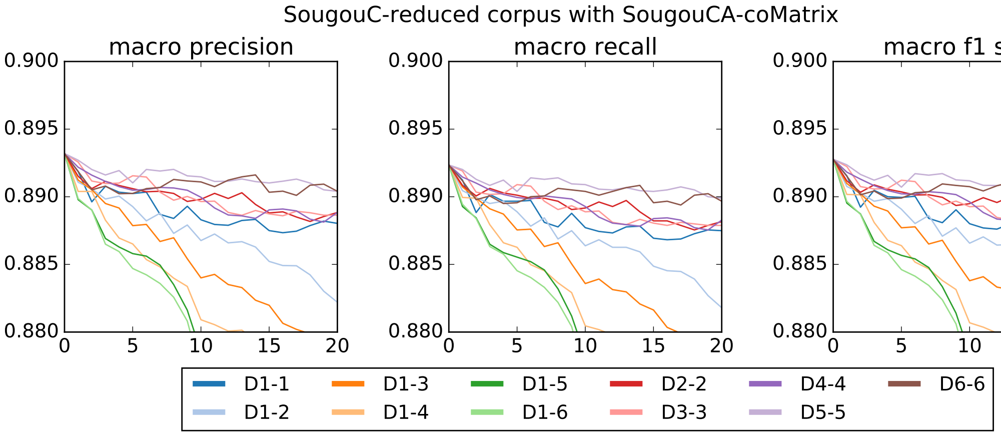


Figure 5.13: Comparison on Sougou classification corpus with character co-occurrence similarity based on SougouCA news.

5.2.4 Discussion

In the experiments, the proposed method obtained promising experimental results with two classification corpora and two external corpora which used as background knowledge. The results indicate that the background knowledge from additional corpora bring the positive impact to the performance. Specially, the proposed method obtained the better performance with the imbalanced corpus, such as Fudan corpus. This indicates that this method can complement the information for the training and test texts when the corpus

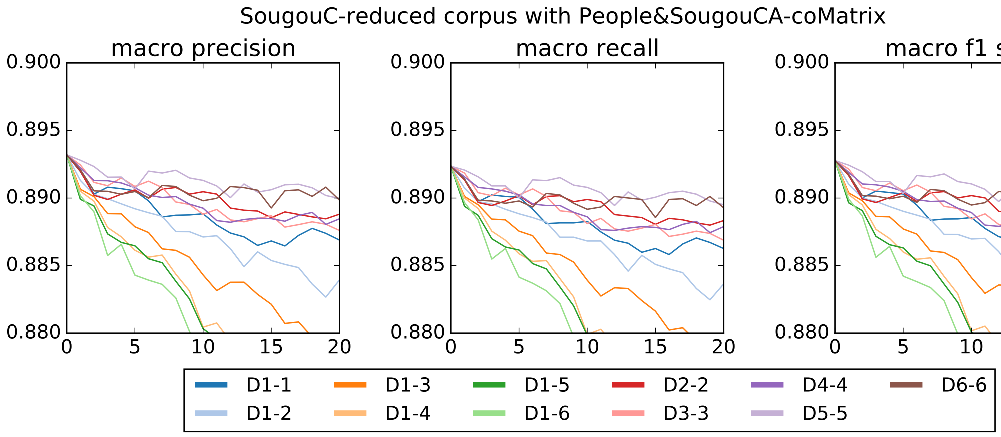


Figure 5.14: Comparison on Sougou classification corpus with character co-occurrence similarity based on People's daily news and SougouCA news.

is imbalanced.

According to the results, using less key characters can improve the performance. The improvement can benefit from a little similarities features based on the background knowledge. This indicates that these similarities are the efficient features for representing texts. The difference of this method from the method based on Baidu Baike is that we do not select the most related characters from the top 20 character further. Because, most words consist of two or more characters in Chinese language commonly. According to the results, when setting the distance larger than 1, the performance can improve. This indicates that words can provide important information. With Sougou corpus, a balanced corpus, the proposed method do not perform well.

Comparing the different background knowledge, using the People's daily news obtained better results with these two classification corpus in the experiments. This indicates that the different background knowledge can impact the results.

5.3 Summary

In this chapter, we evaluated the performance of the proposed methods with many groups of experiments.

The proposed method based on Baidu Baike was evaluated with Fudan text classification corpus. The top 3 keywords of 20 categories were selected by using CTF-ICF algorithm. Totally, 79 articles were downloaded from Baidu Baike using the category names and the keywords. The proposed method added the 79 similarities between the texts and the articles from Baidu Baike to the feature vectors of the texts. Only benefiting from adding the 79 similarities, the experimental results showed that the proposed method outperforms over the traditional methods with Linear SVM classifier. The macro precision, recall and F1 score were improved with 0.02%, 0.15%, 0.12%, comparing to the traditional method. The proposed method obviously improved the recall for some categories with small size and the macro F_1 score. This indicated that the background knowledge can complement the common information between the training and test texts. The proposed method based on Baidu Baike can be regarded as an alternative method for text classification task.

The proposed method based on the characters co-occurrence was evaluated with Fudan corpus and Sougou corpus. The top 20 key characters of 9 categories were selected for each category. After removing the repetitive characters, totally there were 142 key characters in Fudan corpus and 99 key characters in Sougou corpus. The proposed method added the similarities of character co-occurrence between the texts and the background knowledge to the feature vectors of the texts. By analyzing the experimental results, we could make the following conclusion. First, the effectiveness of Chinese text classification could benefit by characters co-occurrence with the background knowledge. Second, the different background knowledge could impact the results. Third, the characters co-occurrence of the different distances between characters could impact the results. When setting the distance larger than 1, it can improve the results. Fourth, the top N key characters played important roles in the proposed method.

Chapter 6

Contribution and Recommendation

6.1 Summary of Text Classification

In this thesis, our work focuses on Chinese text classification task. The background knowledge is used to enrich text representation.

According to the previous researches, the essential problem of text classification is the lack of information, especially for the imbalanced dataset. To solve this problem, we explore the background knowledge to predict class labels for the texts.

The motivation for exploiting background knowledge in text classification is attributed to two reasons. First, more information can make more reasonable classification. Second, people have the basic concepts and general knowledge in their mind, however, the traditional corpora/datasets are some kinds of special case which lack this information. The basic concepts and general knowledge is the background knowledge in our daily life.

Text representation is the fundamental step in text classification task, in which a text could be represented by a set of features. The features play the important roles for training classifiers and predicting class labels. Most of the previous studies focused on enriching text representation to address text classification task. However, the traditional classification methods with VSM (Vector Space Model) only studied intensively on the words and their relationship in some specific corpus/dataset.

In this thesis, we proposed the idea of using background knowledge, which could complement the information for the texts and improve the classifiers. This study is based on Baidu Baike and character co-occurrence. Baidu Baike is an online Chinese encyclopedia which is similar to Wikipedia and widely used by Chinese speakers to learn the basic concepts and general knowledge. The specific articles were downloaded from Baidu Baike, which were ragarded as the background knowledge. The similarities between the texts and the background knowledge were added to the text features for complementing the information. And then, the SVM classifier was used to train model and predict class labels with the enriched text representation. Two additional corpora, People's Daily news and Sougou news, were used for extracting the information of character co-occurrence. From the additional corpora, the character co-occurrence is computed for some key characters, which was used as the background knowledge. These key characters were the most related to their categories. For each text, we also computed the character co-occurrence for itself. The similarities of the character co-occurrence between the texts and the background knowledge were added to the text features for complementing the information. And then, the SVM classifier was used to train model and predict class labels with the enriched text representation.

To decide which articles were download from Baidu Baike, we proposed the CTF-ICF algorithm to select the keywords which are the most related to their categories. This method weights the features/words based on their distribution in the categories. It is similar to TF-IDF algorithm, but the CTF-ICF algorithm works on the category level to represent the categories, while TF-IDF algorithm works on the text level to represent the texts.

The method based on Baidu Baike was evaluated on FUDAN corpus, which includes including 19637 texts within 20 categories. And FUDAN corpus is an imbalanced corpus, in which the different categories have the different numbers of texts. At the beginning, each text was represented by the vector of TF-IDF values. For each text, The dimension of the feature vector was around 350,000 features. Then, the top 3 keywords of 20 categories are selected by using CTF-ICF algorithm, and their articles were downloaded from Baidu Baike. With the category names, 20 articles are downloaded from Baidu Baike. Totally, 79 articles are used as the background knowledge, because one keywords do not have its article in Baidu Baike. The 79 similarities between the texts and the articles from

Baidu Baike were added to the text features. Only Benefiting from the 79 similarities, the experimental results show that the proposed method outperforms over the traditional methods, and obviously improves the recall F1 score for some small categories. This indicated that the background knowledge can complement the common information for the training and test texts. Therefore, the proposed method based on Baidu Baike can be regarded as an alternative method for the text classification task. we also investigated the performance with the different proportion of training and test texts.

The method based on character co-occurrence was evaluated on FUDAN corpus and Sougou corpus. Sougou corpus is a balanced corpus, including 17910 within 9 categories, in which the categories have the same number of texts. At the beginning, each text was represented by the vector of TF-IDF values. Then, the top 20 key characters of 9 categories are selected by combining the texts in each category and weighting the characters with TF-IDF algorithm. After removing the repetitive characters, totally there are 142 characters in Fudan corpus and 99 in Sougou corpus. The proposed method added the similarities of character co-occurrence between the texts and the background knowledge to text features. The results showed that the proposed method improved the performance with the imbalance corpus. This indicated that the method based on character co-occurrence can complement the information between the training and test texts.

With these two proposed methods, we investigated the impact of the background knowledge on text classification. The experiments conducted with two classification corpus, the online encyclopedia, and two additional corpus. The CTF-ICF algorithm was proposed to select the keywords. And the key characters were selected by combining the texts in each categories and using TF-IDF algorithm. According the results, the proposed methods could improve the effectiveness for text classification with the imbalanced corpus.

6.2 Future Directions

We complete this thesis with some discussions about the future directions for text classification and other ideas.

First, the features are the core for text classification with machine learning algorithm to complete the task. In the past, the different kinds of features are the handcraft features.

Recently, the deep learning method becomes popular, because it can learn the features for the examples. The good features can covert the linearly inseparable problem to the linearly separable problem. Therefore, in the future work, we will focus on learning the features in text classification task with the deep learning method. With the learned features, many the machine learning methods can be used to finish the task. In a specific task, the key point is that how to preprocess data for the deep learning method. And building the deep network and modifying hyper-parameter for learning good features are also the key points in practice.

Second, we will focus on selecting features and weighting features for the handcraft features. Selecting features means filtering out the features which are less related to the examples. Weighting features means assigning the high weight to the features which are more related to the example. In deed, they are the same problem. If weighting a feature with 0, weighting features is the same as selecting features. By weighting the features, it can filter out the noise, reduce the dimension of features and improve the performance.

Third, we will explore the new methods about use the external/universal information to enrich the text representation.

To summarize, the first direction tries to find the good features; the second direction tries to make the features better; the third direction tries to complement the features.

Additionally, the research findings and the papers are the direction, instruction and solution for the business, industry, medicine, etc. Therefore, we will also pay attention to the problems/requirements from our daily life, which can be resolved with information and intelligent technologies, such as the machine learning methods.

Bibliography

[1] C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

[2] F. Alías, X. Sevillano, J. Socorǿ, and X. Gonzalvo. Towards high-quality next-generation text-to-speech synthesis: A multidomain approach by automatic domain classification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(7):1340–1354, Sept 2008.

[3] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*, chapter 6. O'Reilly Media, Inc., 1st edition, 2009.

[4] F. Bravo-Marquez, M. Mendoza, and B. Poblete. Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, 69:86–99, 2014.

[5] K. Çelik and T. Güngör. A comprehensive analysis of using semantic information in text categorization. In *Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on*, pages 1–5. IEEE, 2013.

[6] P.-C. Chang, M. Galley, and C. D. Manning. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation*, pages 224–232. Association for Computational Linguistics, 2008.

[7] E. Chen, Y. Lin, H. Xiong, Q. Luo, and H. Ma. Exploiting probabilistic topic models to improve ext categorization. *Information Processing & Management*, 47:202–214, March 2011.

[8] T. F. Covões and E. R. Hruschka. Towards improving cluster-based feature selection with a simplified silhouette filter. *Information Sciences*, 181(18):3766–3782, 2011.

[9] F. Enríquez, F. L. Cruz, F. J. Ortega, C. G Vallejo, and J. A. Troyano. A comparative study of classifier combination applied to nlp tasks. *Information Fusion*, 14(3):255–267, July 2013.

[10] S. Feldman, M. A. Marin, M. Ostendorf, and M. R. Gupta. Part-of-speech histograms for genre classification of text. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4781–4784. IEEE, April 2009.

[11] F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. Meira Jr. Word co-occurrence features for text classification. *Information Systems*, 36(5):843–858, 2011.

[12] H. Guan, J. Zhou, B. Xiao, M. Guo, and T. Yang. Fast dimension reduction for document classification based on imprecise spectrum analysis. *Information Sciences*, 222:147–162, Feb. 2013.

[13] M. Hrala and P. Král. Evaluation of the document classification approaches. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 877–885. Springer, 2013.

[14] R. Hu, B. M. Namee, and S. J. Delany. Active learning for text classification with reusability. *Expert Systems with Applications*, 45:438–449, 2016.

[15] I. Hwang, H. M. Park, and J. H. Chang. Ensemble of deep neural networks using acoustic environment classification for statistical model-based voice activity detection. *Computer Speech & Language*, 38:1–12, 2016.

[16] E. L. Iglesias, A. Seara Vieira, and L. Borrajo. An hmm-based over-sampling technique to improve text classification. *Expert Systems with Applications*, 40(18):7184–7192, Dec. 2013.

[17] L. Jiang, C. Li, S. Wang, and L. Zhang. Deep feature weighting for naive bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, 52:26–39, 2016.

[18] S. Jiang, G. Pang, M. Wu, and L. Kuang. An improved k-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, 39(1):1503–1509, Jan. 2012.

[19] K. Kim, B. suk Chung, Y. Choi, S. Lee, J. Y. Jung, and J. Park. Language independent semantic kernels for short-text classification. *Expert Systems with Applications*, 41:735–743, Feb. 2014.

[20] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273, 2015.

[21] C. Lioma and R. Blanco. Part of speech based term weighting for information retrieval. In *Advances in Information Retrieval*, volume 5478, pages 412–423. Springer Berlin Heidelberg, 2009.

[22] S. Maldonado, R. Weber, and J. Basak. Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences*, 181(1):115–128, 2011.

[23] S. Maldonado, R. Weber, and F. Famili. Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences*, 286:228–246, Dec. 2014.

[24] A. Moschitti. Kernel methods, syntax and semantics for relational text categorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 253–262. ACM, 2008.

[25] A. Moschitti and R. Basili. Complex linguistic features for text classification: A comprehensive study. In *European Conference on Information Retrieval*, pages 181–196. Springer, 2004.

[26] A. Onan, S. Korukoğlu, and H. Bulut. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57:232–247, 2016.

[27] G. Pang and S. Jiang. A generalized cluster centroid based classifier for text categorization. *Information Processing & Management*, 49(2):576–586, March 2013.

[28] Y. Qian, Y. Liang, M. Li, G. Feng, and X. Shi. A resampling ensemble algorithm for classification of imbalance problems. *Neurocomputing*, 143:57–67, Nov 2014.

[29] C. Quan and F. Ren. A blog emotion corpus for emotional expression analysis in chinese. *Computer Speech & Language*, 24(4):726–749, 2010.

[30] C. Quan and F. Ren. Unsupervised product feature extraction for feature-oriented opinion determination. *Information Sciences*, 272:16–28, 2014.

[31] M. Rafi, S. Hassan, and M. S. Shaikh. Content-based text categorization using wikitology. *arXiv preprint arXiv:1208.3623*, 2012.

[32] F. Ren. From cloud computing to language engineering, affective computing and advanced intelligence. *International Journal of Advanced Intelligence*, 2(1):1–14, 2010.

[33] F. Ren and C. Li. Hybrid chinese text classification approach using general knowledge from baidu baike. *IEEJ Transactions on Electrical and Electronic Engineering*, 2016.

[34] F. Ren and M. G. Sohrab. Class-indexing-based term weighting for automatic text classification. *Information Sciences*, 238(0):109–125, July 2013.

[35] L. Rocha, F. Mourão, H. Mota, T. Salles, M. A. Gonçalves, and W. Meira Jr. Temporal contexts: Effective text classification in evolving document collections. *Information Systems*, 38(3):388–409, 2013.

[36] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera. Smote-ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203, 2015.

[37] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[38] C. Shang, M. Li, S. Feng, Q. Jiang, and J. Fan. Feature selection via maximizing global information gain for text classification. *Knowledge-Based Systems*, 54:298–309, Dec. 2013.

[39] K. Shi, J. He, H. tao Liu, N. tong Zhang, and W. Song. Efficient text classification method based on improved term reduction and term weighting. *The Journal of China Universities of Posts and Telecommunications*, 18:131–135, Sep. 2011.

[40] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández. Syntactic dependency-based n-grams as classification features. In *Mexican International Conference on Artificial Intelligence*, pages 1–11. Springer, 2012.

[41] D. Torunoğlu, G. Telseren, Ö. Sağtürk, and M. C. Ganiz. Wikipedia based semantic smoothing for twitter sentiment classification. In *Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on*, pages 1–5. IEEE, 2013.

[42] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of*

the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 173–180. Association for Computational Linguistics, 2003.

[43] K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

[44] B. Trstenjak, S. Mikac, and D. Donko. Knn with tf-idf based framework for text categorization. *Procedia Engineering*, 69:1356–1364, 2014.

[45] C. Tsai and C. Chang. Svois: Support vector oriented instance selection for text classification. *Information Systems*, 38:1070–1083, Nov. 2013.

[46] C. Tsai, Z. Chen, and S. Ke. Evolutionary instance selection for text classification. *Journal of Systems and Software*, 90:104–113, April 2014.

[47] A. K. Uysal and S. Gunal. Text classification using genetic algorithm oriented latent semantic features. *Expert Systems with Applications*, 41:5938–5947, Oct. 2014.

[48] C. Wang, M. Zhang, S. Ma, and L. Ru. Automatic online news issue construction in web environment. In *Proceedings of the 17th international conference on World Wide Web*, pages 457–466. ACM, 2008.

[49] D. Wang, J. Wu, H. Zhang, K. Xu, and M. Lin. Towards enhancing centroid classifier for text classification—a border-instance approach. *Neurocomputing*, 101:299–308, Feb. 2013.

[50] P. Wang, J. Hu, H. Zeng, and Z. Chen. Using wikipedia knowledge to improve text classification. *Knowledge and Information Systems*, 19(3):265–281, 2009.

[51] S. Wang, D. Li, L. Zhao, and J. Zhang. Sample cutting method for imbalanced text sentiment classification based on brc. *Knowledge-Based Systems*, 37:451–461, Jan. 2013.

[52] R. Xia, C. Zong, and S. Li. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181(6):1138–1152, 2011.

[53] B. Zhang, A. Marin, B. Hutchinson, and M. Ostendorf. Learning phrase patterns for text classification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(6):1180–1189, June 2013.

[54] H. Zhang and G. Zhong. Improving short text classification by learning vector representations of both words and hidden topics. *Knowledge-Based Systems*, 102:76–86, 2016.

[55] L. Zhang, L. Jiang, C. Li, and G. Kong. Two feature weighting approaches for naive bayes text classifiers. *Knowledge-Based Systems*, 100:137–144, 2016.