國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

# 使用輔助向量的雙邊特徵分群以改善中文新聞的立場偵測分類

# Two-side Feature Clustering Using Auxiliary Vector for Improving Stance Classification on Chinese Newspaper

陳韋銘

Wei-Ming Chen

指導教授：林守德 博士

Advisor: Shou-De Lin, Ph.D.

中華民國 105 年 2 月

February 2016

# 誌謝

能夠完成這篇論文，要特別感謝我的指導教授林守德老師，給我思考與選擇碩士論文題目的權利，也在研究的過程中不遺餘力地給予我指導與建議。感謝李政德博士與古倫維博士兩位口試委員，給予我重要且實際的回饋。感謝實驗室的同儕們，許晉齊、黃彥樺、黃宇陽、李揚、陳彥齊、郭冠呈同學，以及網路探勘與資訊檢索實驗室的施亭屹、姜俊宇同學，在我充滿疑問時總是樂意與我討論，讓我受益良多。感謝實驗室學弟冉昱與謝明倫，協助我撰寫網路爬蟲以蒐集新聞資料。最後想要感謝詹婞筑、江明峰、黃禹力、郭子熒、邱子軒、廖柏鈞、施亭屹、許晉齊、陳彥齊、黃彥樺以及王邦丞同學，協助我標記千餘篇新聞的立場，我方能有資料進行研究。

# 中文摘要

為了紓解媒體偏頗以及閱聽者選擇性偏好的現象，本篇論文專注於發展一智慧程式，用以分辨中文爭議性議題新聞之立場。我們提出一個簡單且有效率的方法，能夠考量無標記新聞資料庫的資訊、以及訓練資料之資訊，以合併相似的特徵。在我們提出的方法中，特徵會先根據初始訓練過程被分為兩邊，接著使用word2vec工具為每一個特徵產生輔助向量，最後使用高速的社群偵測演算法將意義上相近的特徵合併。實驗結果顯示，在大多數的情況下，我們提出的解決方案比直接使用原始特徵、以及使用常見的降維演算法還要好。

關鍵字：立場偵測、中文新聞立場偵測、特徵合併、自然語言處理、機器學習。

# ABSTRACT

In order to relieve media bias problem and selective preference problem, we aim at developing an intelligent system to classify the stance of Chinese news article on several controversial topics. We proposed a simple and efficient approach which can incorporate the information of unlabeled news corpus and the information of training data to merge similar features. In our approach, features were divided into two sides according to initial training process, and word2vec tool was utilized to produce auxiliary vectors for each feature. Finally, fast community detection algorithm was applied for clustering similar features. Experimental results show that our approach outperforms raw features and common dimensionality reduction techniques in most cases.

Keywords: stance classification, stance classification on Chinese newspaper, feature clustering, natural language processing, machine learning.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1 Introduction

## 1.1 Motivation and Overview

Online news website has become prevailing in past years, and there have been lots of readers taking online news as their main news source because it is free and fast. However, not only media sometimes expresses its own ideological stance (media bias problem), but also readers tend to select what they want to read (selective preference problem). It is possible that news readers usually absorb information only from one standpoint, and this phenomenon may lead to people misunderstanding important issues or even raising confliction between the people. Changing the stance of media is difficult, but altering the way of displaying news may relieve this problem. Our research aims at building an intelligent system which can classify the stance of news in controversial topics. With this system, we can divide news into several groups with different stance, and then automatically deliver news with different stance to readers. We believe in this way, readers can receive multi-viewpoint information more easily to understand important issues in their countries.

We collected news articles of 7 controversial topics in Taiwan from 7 online new websites, and we built a website to annotate the stance of 1177 news articles for 4 topics. After labeling stance of news articles, we extract informative features from the content of news article, and we discovered that neutral word in dependency features plays an important role for classifying stance in news domain. In order to improve performance of stance classification and to reduce dimensionality to avoid overfitting in such small data condition, we propose a simple and efficient approach which can not only incorporate information from unlabeled news corpus to merge features but also consider the label information of training data to avoid merging features from different stance. Experimental results show that our approach outperforms original raw features and other common dimensionality reduction techniques in most of cases.

## 1.2 Problem Formulation

**Definition 1. Stance Classification Problem.**

Given an article $d$ of certain controversial topic, and a stance statement $s$ related to this topic, the goal is to answer whether this article $d$ “$agree$”, “$oppose$” $or$ “$is\ neutral\ to$” the stance statement $s$.

We take an example for illustration. Suppose $d$ is an article persuading people in Taiwan to sign Cross-Trait Service Trade Agreement with China government, and the stance statement $s$ is “Taiwan should sign Cross-Trait Service Trade Agreement with China”, then our goal is to create an intelligent computer program to answer “$agree$”.

In our research, we conduct stance classification on Chinese news article, and we simplified the stance classification problem to a binary classification problem to only answer “$agree$” or "$oppose$". Details will be shown in Chapter 3.

# Chapter 2 Related Work

Classifying the stance of a post is a relatively new opinion mining problem, and there has been a growing body of works trying to tackle this challenging task. As stated in [5], previous works mainly cover three different kinds of settings: (1) company internal discussion [12] (2) congressional debates [9][10][11] (3) online social and political debate forum [1][3][4][5][6][7][8]. Debates in online forum differ from debates in congress and in company, online debaters often using emotional and irony language to express their opinions, and they also have strong personal belief in some of topics. These properties make debate-side stance classification more challenging. However, most of works are in debate-side stance classification due to the growing available data in popular online debate forums[1][2][3].

Instead of covering above three different settings, our work aims at doing stance classification in online Chinese news article. We believe that correctly classifying stance of news article can help deliver news with different viewpoints.

Most of previous works extracted discriminative features from posts, and utilized supervised machine learning techniques to train a stance classifier. Bag-of-word, n-grams, statistics of repeated punctuations, cue words, and quotations features were used as basic features. Besides, polarity-target pair features, full pair features and other variants of dependency features were generated from dependency parsing tree as advanced text features [1][2][3][4][6][11][14]. Frame semantic features were also created to enhance the prediction performance [1][8]. However, bag-of-word feature has

[1] http://www.convinceme.net/
[2] http://www.createdebate.com/
[3] http://www.4forums.com/

been shown as a strong baseline, and most of advanced text features can only improve around 3-7% accuracy.

Besides above text features, recent works have shown that utilizing debate-specific information can largely improve the prediction performance. Debate-specific information is 3-folds: (1) thread of debate posts: Online debaters reply or rebut to previous debater's posts, which forms a tree-like structure in a debate thread. Because 80%-90% responses are against previous posts, User-interaction Constraints were used to enhance prediction performance in [6][7]. (2) Author information: Debaters usually held same stance in a debate thread or even in whole website, and debaters with similar stances in domain A can be a message to having a similar stance in domain B. In [5][7], the researchers employed these information as Author Constraints to achieve better prediction. (3) Rebuttal links: Debater can explicitly claim to rebut previous post in some online forums. Accompanying with author information, a post-to-post graph was built [5], where each node represents a post, positive edges represent two posts with same author and negative edges represent explicit rebuttal links. This graph-based approach has been shown to be effective in stance classification.

Although these debate-specific properties provide plentiful information, news articles are not dialogic and the authors of news may not express their stance explicitly while considering the ethics of journalism. Lacking of debate-specific information, our work aims at improving prediction performance from only text information.

# Chapter 3 Dataset

## 3.1 Data Collection

We crawled around 240,000 pieces of news and comments from 7 online news websites in Taiwan[4][5][6][7][8][9][10]. 7852 articles of 7 controversial topics were collected if the article contains topical keywords for more than or equal to 3 times. Statistics of articles before annotation and the keywords of each topic are also listed in Table 1.

| Topic | Keywords | #Doc |
|---|---|---|
| 核四興建案爭議 | 核一、核二、核三、核四、龍門核能發電廠、第四核能發電廠、第四核電廠、核能第四發電廠、反核、核能、核電、廢核、擁核、核廢料、核災、核安 | 3206 |
| 海峽兩岸服務貿易協議爭議 | 服貿、服務貿易 | 2233 |
| 自由經濟示範區爭議 | 自由經濟示範區、示範區、自經區 | 375 |
| 台灣進口美國牛肉爭議 | 美牛、美國牛肉 | 384 |
| 多元成家方案爭議 | 多元成家、婚姻平權、同性婚姻、伴侶盟、伴侶權益推動、同性戀婚姻、同性戀成家、同志婚姻、同志成家、守護家庭、護家盟 | 526 |
| 台灣基本薪資調漲爭議 | 基本工資、基本薪資、最低薪資、最低工資" | 914 |
| 美麗灣渡假村開發案爭議 | 美麗灣、杉原海岸 | 214 |

Table 1. Number of news articles and the keywords of each controversial topic

[4] 苦勞網: http://www.coolloud.org.tw/
[5] 三立新聞網: http://www.setn.com/
[6] 自由時報電子報: http://www.ltn.com.tw/
[7] 風傳媒: http://www.storm.mg/
[8] 聯合新聞網: http://udn.com/news/index
[9] 關鍵評論網: http://www.thenewslens.com/
[10] 公視新聞議題中心: http://pnn.pts.org.tw/main/

## 3.2 Data Annotation

After crawling articles from online news website, we built an annotation website to annotate the stance of each selected article. Due to time constraints, only some of articles were selected to label the stance. For each selected article, there are 3 questions to answer (1) Format validness: Are there lots of encoding errors or advertisement text in article, or is there any paragraph missing in this article (*valid/invalid*)? (2) Relevance: Is this article highly relevant to this controversial topic (*relevant/irrelevant*)? (3) Stance: What is the overall stance of this article toward the statement (*agree/oppose/neutral*)?

Each article was annotated by two different annotators, and it was removed when being annotated as *invalid* or *irrelevant*. If one article was firstly annotated as *valid*(*relevant*) but was secondly annotated as *invalid*(*irrelevant*), then this article will be annotated the third time to decide the final answer. The flow of annotation is shown in Figure 1.

After removing *invalid* and *irrelevant* articles, we calculated stance scores to determine stance of each article. Stance *agree, neutral, oppose* have scores +1, 0, and -1 respectively, and the final stance of an article is *agree*, *neutral* and *oppose* when sum of scores is >0, =0 and <0 respectively.

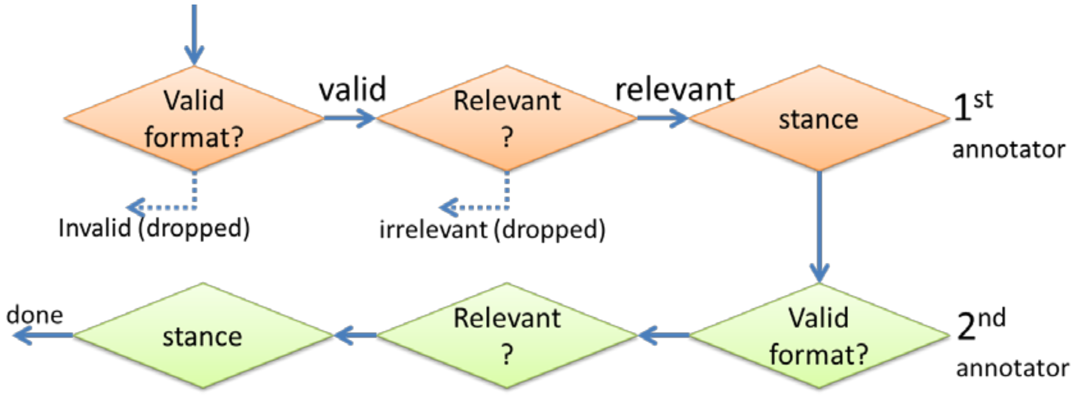


Figure 1. The flow of annotation

## 3.3 Data Observation

We have 739, 116, 128, 194 news articles in topic 1, 2, 3 and 4 respectively and inter-annotator agreement coefficient (Krippendorff's alpha coefficient α [15]) is shown in Table 2.

We can observe that there is a major stance class in most of topics, which means media may be biased toward certain stance, which could lead audience to only absorb information from one side. As a consequence, it is crucial to build an intelligent system to detect the stance of news article so that we can relieve this problem by delivering news article with different stances.

| topic | stance statement | agree | neutral | oppose | total | $\alpha$ |
|---|---|---|---|---|---|---|
| 1 | 應簽訂服務貿易協議 | 201 (27.2%) | 145 (19.6%) | 393 (53.2%) | 739 | 0.616 |
| 2 | 應簽訂自由經濟示範區條例 | 49 (42.2%) | 13 (11.2%) | 54 (46.6%) | 116 | 0.612 |
| 3 | 台灣應進口美國牛肉 | 29 (22.7%) | 19 (14.8%) | 80 (62.5%) | 128 | 0.594 |
| 4 | 台灣不應調漲基本薪資 | 25 (12.9%) | 43 (22.2%) | 126 (64.9%) | 194 | 0.716 |
| Total | | 304 (25.8%) | 220 (18.7%) | 653 (55.5%) | 1177 | 0.654 |

Table 2. Number of news articles in each stance and Krippendorff's alpha coefficients for each topic.

However, because there are too few articles with *agree* or *neutral* stance for training a classifier, we merge *neutral* stance to *agree* stance to simplify original task. *Agree* stance is actually the combination of *agree* and *neutral* stance but we still call it *agree* stance for convenience. In such way, we can have a data-balanced binary classification task. Besides, for better presentation, we abbreviated topic 1 to 4 as 服貿, 自經, 美牛, 基薪 in the rest of paper. The summary of our data is shown in Table 3.

| topic | stance statement | agree | oppose | total |
|---|---|---|---|---|
| 1 | 應簽訂服務貿易協議（**服貿**） | 346 (46.8%) | 393 (53.2%) | 739 |
| 2 | 應簽訂自由經濟示範區條例（**自經**） | 62 (53.4%) | 54 (46.6%) | 116 |
| 3 | 台灣應進口美國牛肉（**美牛**） | 48 (37.5%) | 80 (62.5%) | 128 |
| 4 | 台灣不應調漲基本薪資（**基薪**） | 68 (35.1%) | 126 (64.9%) | 194 |
| Total | | 524 (44.5%) | 653 (55.5%) | 1177 |

Table 3. Number of news articles in each stance after merging *neutral* stance to *agree* stance, and the abbreviation of each topic.

# Chapter 4 Methodology

In our research, we utilized supervised learning approach to reach our goal, and we selected logistic regression classifier (maximum entropy classifier) as our classifier, which is shown to be effective in many sentiment prediction task.

In next three sections, we firstly demonstrated how to extract informative features from news articles, and secondly we reviewed several common dimensionality reduction techniques as our baseline. Finally, we proposed a feature merging approach to improve performance of stance classification, in which information of unlabeled news corpus and information of training data were considered.

## 4.1 Feature Extraction

### 4.1.1 Word-based Feature

**Bag-Of-Word**

Bag-of-word feature is one of the simplest but useful features in many natural language tasks. We simply count the occurrence of each word in a document as our feature, but we only allow the words with NN(noun), NR(proper noun), AD(adverb), VV(verb), VA(adjective) and JJ(other noun-modifier) part-of-speech tags.

**N-grams**

N-gram is originally used to estimate the likelihood of a sentence by conditional probability. Here we use concept of N-grams to extract features from document. We count the occurrence of consecutive 2 and 3 words in a document as our feature, so we call them Bi-Word and Tri-Word feature respectively. Similar to bag-of-word feature, we only allow the word sequence where the part-of-speech tag of at least one word is NN, NR, AD, VV, VA, and JJ, and none of them is PU(punctuation).

Besides, we take **BOW**, **BiWord**, **TriWord** as the abbreviation of bag-of-word feature, Bi-Word feature and Tri-Word feature respectively.

### 4.1.2 **Dependency-based Feature**

Dependency is the notion that words are connected to each other by directed links, and verb is usually taken as the center of clause structure. Dependency parsing has become an important natural language task, because its result provides dependency information between word and word in a sentence, which is very useful for many natural language tasks.

In order to illustrate how we use dependency relation as features, we first formally define some necessary notation.

**Definition 2. Dependency Relation**

A dependency relation $r = (w_h, w_d, t_h, t_j, d)$ is a direct link from $w_h$ to $w_d$ where $w_h$ is *head word* (usually verb), $w_d$ is *dependent word*, and $d \in D$ is *type of dependency relation*. The type of dependency relation indicates the syntactic relation between $w_h$ and $w_d$, and the set of all possible dependency relation types are manually defined in advance. Besides, the part-of-speech (POS) tags of $w_h$ and $w_d$, $t_h$ and $t_d$ are usually attached on the dependency relation at same time.

The meaning of each type of dependency relation in Stanford Chinese dependency parser is defined in [16]. Besides, in researches of opinion mining, head word is usually viewed as opinion word, and the dependent words is viewed as (opinion-)target word.

**Definition 3. Dependency Tree**

A dependency tree $T = \{ r_1, r_2 \ldots r_n \}$ is a set of dependency relations extracted from one sentence and be formed as tree structure. Nodes in dependency tree are words in the sentence and edges are dependency relations.

There are several variants of representation form of dependency features, and we implement *Full Pair* and *Polarity-Target Pair* as our features. Before illustrating how we extract features from dependency relation, we firstly deal with negation words in a sentence.

**Dealing with negation word**

We count the number of negation dependency relation which connected to head word to decide the negation sign of this head word. If the number of negation relation is odd then negation sign is -1, otherwise it is +1. A real example of sentence “我不支持服貿” is shown in Figure 2. In Stanford Chinese dependency parser, *neg* denotes the negation dependency relation.

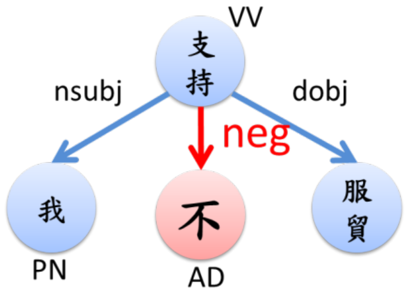


Figure 2. The dependency tree of “我不支持服貿”

**Full Pair Dependency Feature**

In this form, we directly take $w_h$, $w_d$ and negation sign into account, so we name it as Full Pair dependency feature. The representation form is

$(\text{negation\_sign}, w_h, w_d)$, and the features is the count of each form of Full Pair dependency relation in a document.

**Polarity-Target Pair Dependency Feature**

As a result of low generalization ability of Full Pair representation form, we replaced head word to polarity by inquiring sentiment lexicon. We used the core version of National Taiwan University Sentiment Dictionary (NTUSD) [17] as our sentiment lexicon. The polarity value is +1/-1 if the word has positive or negative sentiment, otherwise it is 0. The representation form is $(\text{negation\_sign} \times \text{polarity}, w_d)$. Similar to Full Pair form, we count each form of dependency relation as features.

We take sentence “我支持服貿” as example, the dependency tree of this sentence and the representation form of Full Pair and Polarity-Target Pair dependency feature are shown in Figure 3.

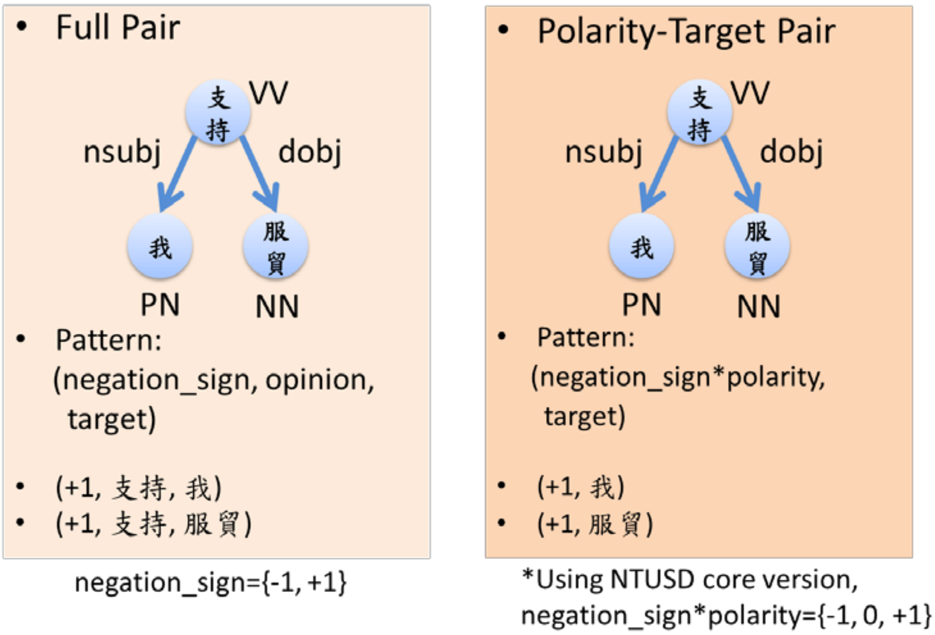


Figure 3. An example of Full Pair dependency feature (left) and Polarity-Target Pair dependency feature (right).

Let **Full** denote Full Pair dependency feature and **PT** denote Polarity-Target Pair dependency feature.

**Extracting Dependency Relation**

In previous work, dependency relation is usually extracted by head word when head word has positive or negative sentiment. However, we found this approach may lose the information of dependency relation containing neutral word, which can be shown to be very important in news domain. As a consequence, we use the part-of-speech tags of two words as filtering criteria, and the allowed types are *amod*, *dobj* and *nsubj*. *Amod* denotes *adjectival modifier* relation, *dobj* denotes *direct object* relation and *nsubj* denotes *nominal subject* relation.

For convenience, we abbreviate dependency features PT, Full extracted by sentiment lexicon as **PT_SB** and **Full_SB** respectively. Similarly, **PT_TB** and **Full_TB** are the abbreviation of dependency features **PT** and **Full** extracted by part-of-speech tags.

## 4.2 Dimensionality Reduction Techniques

### 4.2.1 Feature selection based approach

There usually exists a measurement for measuring how important or how discriminative of certain feature is in feature selection based approaches. Here we use Chi-Square (Chi), L1-norm (L1), Random Forest (RF), Recursive Feature Elimination (RFE), and Recursive Feature Elimination with Cross-Validation (RFECV) as our baselines.

**Chi-square**

Chi-square statistics measure the dependence between stochastic variables, thus we remove features that are most likely to be independent of stance (label).

**L1-norm**

L1-regularization usually leads to sparse coefficients (weights) of linear classifier, and here we use linear Support Vector Machine (SVM) with L1-regularization to do feature selection. The features with smallest absolute value of coefficients will be removed.

**Random Forest**

Random Forest has been a common and useful classifier for many machine learning tasks. We remove the features with smallest feature importance calculated by random forest.

**Recursive Feature Elimination**

Recursive Feature Elimination is a kind of procedure of removing features. In each round, we train a classifier using training data and then remove k features according to some feature importance measurement, such as the chi-square statistics, feature importance in random forest and absolute value of coefficient in linear classifier. We repeat several rounds until reaching terminating criteria.

**Recursive Feature Elimination with Cross-Validation**

In Recursive Feature Elimination with Cross-Validation, we use cross-validation to decide the best feature number, and then run the Recursive Feature Elimination.

### 4.2.2 Low rank approximation

Low rank approximation is another important branch to reduce feature dimension. This kind of approach projects original features to a new lower rank space, thus noise information can be removed. We applied Principal Component Analysis (PCA), which is a very basic and common approach.

**Principal Component Analysis**

Principal component analysis (PCA) is a statistical process to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

## 4.3 Our Proposed Approach

We proposed *Two-side Feature Clustering using Auxiliary Vector* as our approach. In our approach, we first divide features into two sides according to their coefficients in linear classifier, and then divide features into subgroups according to feature types. Second, we generate auxiliary vector for each feature by applying word2vec tool [19][20][21] on unlabeled news corpus. Third, we build feature graph by calculating similarity between features and create edges if the similarity is larger than given threshold. Finally, we run community detection algorithm on feature graph, and then the features in same community are merged into one feature.

In the following four sections, we will illustrate the four steps of our approach in details, and the overview of our approach is shown in Figure 4.

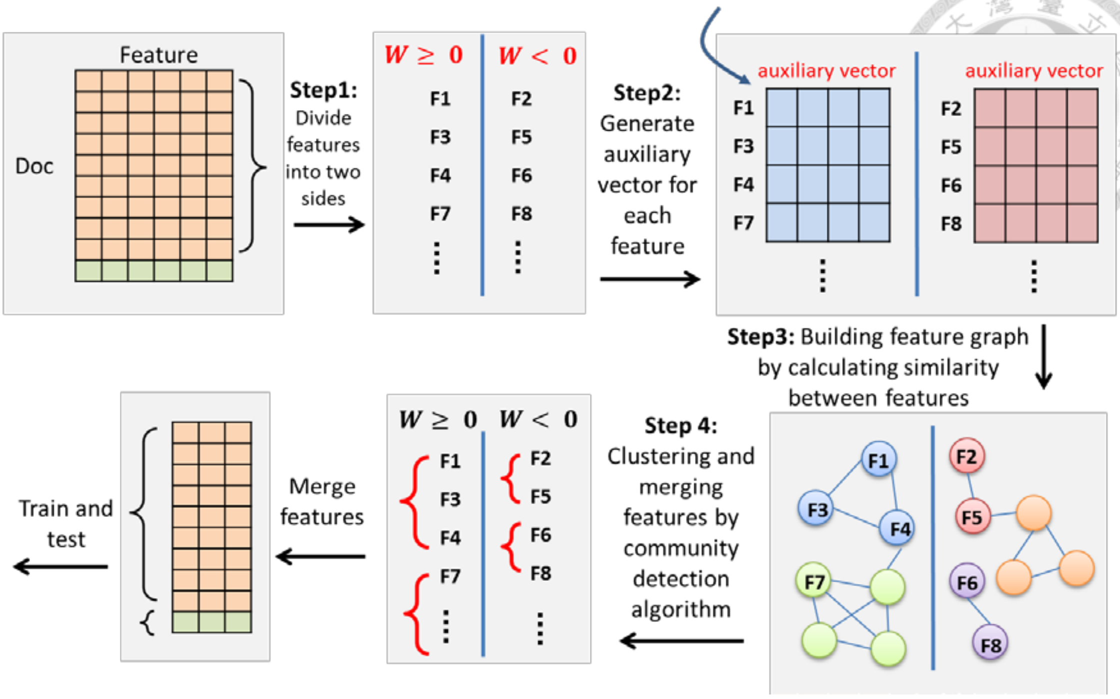

Feature
Doc
Step1: Divide features into two sides
$W \geq 0$
$W < 0$
F1
F3
F4
F7
F2
F5
F6
F8
Step2: Generate auxiliary vector for each feature
auxiliary vector
auxiliary vector
Step3: Building feature graph by calculating similarity between features
Step 4: Clustering and merging features by community detection algorithm
Merge features
Train and test

Figure 4. The overview of our proposed approach.

**Step 1. Divide features into two sides**

In first step, we use logistic regression to train a classifier using training data, thus we have coefficient $\mathrm{w_i}$ of feature $\mathrm{f_i}$. Later we divide features into two sides, $\mathrm{W_+} = \{\mathrm{f_i} | w_i \geq 0\}$ and $\mathrm{W_-} = \{\mathrm{f_i} | w_i < 0\}$, where $\mathrm{W_+}$ denotes the set of features with *agree* or no tendency and $\mathrm{W_-}$ denotes the set of features with *oppose* tendency.

Without this step, performance will decrease because features with different stance tendency may be merged. We will confirm this later in experiments sections. Thus, we have to divide features into two sides to avoid this problem.

**Step 2. Generate auxiliary vector for each feature**

Auxiliary vector should be designed for capturing the similarity between features, such that we can utilize it to measure how similar the two features are to build feature graph in next step. We apply word2vec tool to generate word vector from the unlabeled news corpus in Table 1, and there are totally 7852 news articles containing 4,789,940 words in this corpus. With word vector in hand, we calculate auxiliary vector for each feature by vector addition and scalar multiplication. The details are shown in Table 4.

| **Feature type** | **Pattern** | **Auxiliary vector** | **Group** |
|---|---|---|---|
| BOW | $v_i$ | $v_i$ | BOW |
| Bi-Word | $(v_i,\ v_j)$ | $v_i + v_j$ | BiWord |
| Tri-Word | $(v_i,\ v_j,\ v_k)$ | $v_i + v_j + v_k$ | TriWord |
| Full Pair | $(neg,\ v_i,\ v_j)$ | $neg \times (v_i + v_j)$ | Full |
| Polarity-Target Pair | $(np,\ v_i)$<br>$where\ np = neg \times pol$ | $v_i,\ if\ np = 0$ | PT_Neutral |
| | | $np \times v_i,\ if\ np\ \neq 0$ | PT_PN |

Table 4. The way of auxiliary vector calculation for each type of features, where $\mathrm{v_i}$ denotes the word vector of i-th word.

Because in original works or word2vec tool, it claimed that word vector keeps physical meaning while doing addition and subtraction operation, such as $\mathrm{v_{king}} - v_{men} + v_{woman} \cong v_{queen}$. We utilize this great property to design our auxiliary vector for each kind of features.

For BOW, auxiliary vector is the vector of original word, and for Bi-Word and Tri-Word, the auxiliary vector can be simply calculated from addition of the word vectors in that feature. Besides, for Full Pair form of dependency feature, we add the two word vectors and then we multiply the vector by negation sign, +1 or -1. Finally, for Polarity-Target Pair form of dependency feature, we firstly divide the feature into two groups, one is *PT_Neutral* for $\mathrm{np} = 0$ and another is *PT_PN* for $\mathrm{np} \in \{+1, -1\}$, where $\mathrm{np} = \mathrm{negation\_sign} \times \mathrm{polarity}$. For *PT_Neutral*, auxiliary vector is simply the vector of target word, and for *PT_PN* auxiliary vector is calculated by $\mathrm{np} \times \mathrm{v_i}$.

Moreover, fourth column of Table 4 denotes the group of each kind of features, which means that when we only calculate similarity between the features in same group.

**Step 3. Building feature graph by calculating similarity between features**

With auxiliary vector for each feature, we can build feature-to-feature graph for later clustering features. In this step, we calculate cosine similarity $\mathrm{s_{i,j}} = cosine(u_i, u_j) \in [-1,1]$ between two auxiliary vectors $\mathrm{u_i}$ and $u_j$ for feature $\mathrm{f_i}$ and $\mathrm{f_j}$ in same group. If $\mathrm{s_{i,j}}$ is larger than given threshold, then an edge $\mathrm{e_{i,j}}$ will be added to graph. This is the most time consuming step in our approach, but we can build feature-to-feature graph offline for only one time and cutting several edges when dividing feature into two sides in Step 1.

**Step 4. Clustering and merging features by community detection algorithm**

In final step, we already have feature-to-feature graph, thus we run community detection algorithm to cluster the features in same community. *Community detection* is the way of dividing a network into groups of nodes with dense connections internally and sparser connections between groups. We use Louvain method for community detection, and it is a simple, efficient and effective algorithm which greedily optimizes *Modularity* $\mathrm{Q} = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta_{ij}$ in its procedure, where $\mathrm{A_{ij}} = 1$ indicates there is an edge between node i and node j, $\mathrm{k_i}$ denotes the degree of node i, $\mathrm{\delta_{ij}} = 1$ denotes node i and node j are in same community and m denotes the number of edges in graph. Modularity Q can be viewed as the magnitude of how nodes are connected in community and how nodes do not connected between communities.

After running community detection algorithm on feature graph, we merge the features in same community by simply summing the feature vectors.

# Chapter 5 Experiments

## 5.1 Experimental Settings

In our experiments, we run 10-fold cross validation for 3 random seeds, so there are totally 30 rounds for each experiment. Besides, for each round we further do cross-validation again to search the best parameters of classifier, so actually there are 300*N training and testing procedure when searching parameters, where N is the number of all possible combination of parameters. Besides, we simply take accuracy as our evaluation metric because data is balanced after merging *neutral* stance into *agree* stance. The illustration of experimental setting is shown in Figure 5, where each row is the news article of a topic.

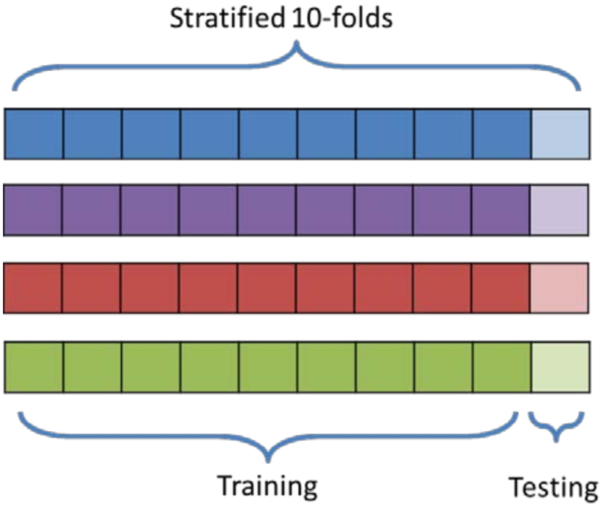
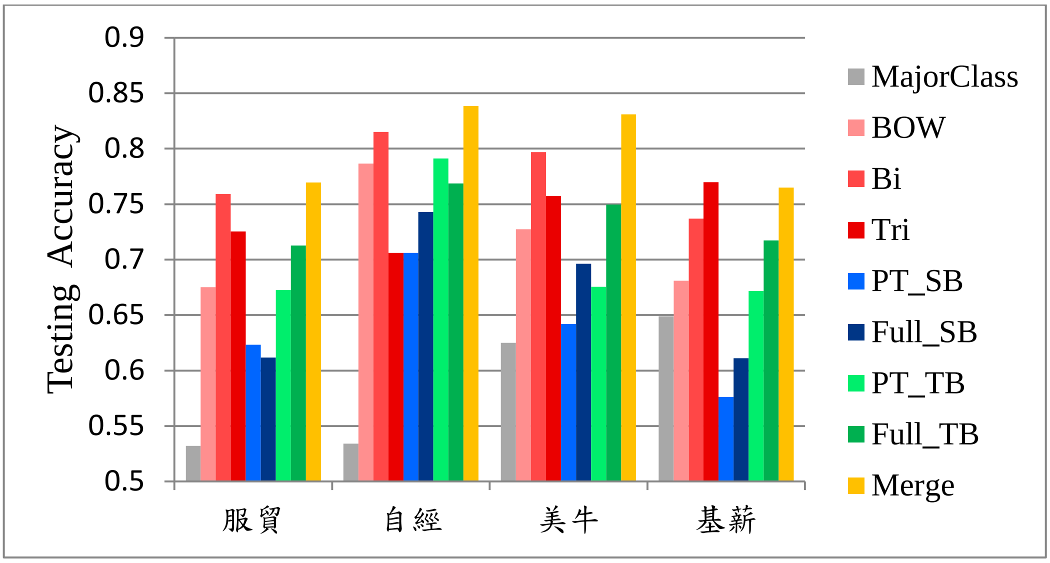


Figure 5. Illustration of evaluation.

## 5.2 Experimental Results

In this section, the performance of each single type of feature and merged feature is shown first, and we discovered that neutral word is the key to do stance classification in news domain. Second, we compare the performance of our proposed approach to the performance of directly clustering feature without dividing features into two sides. The results show that dividing features into two sides is a critical step for merging features. Finally, we compare our approach to other baseline approaches, and the results show that our proposed approach outperformed other baseline in most all of conditions.

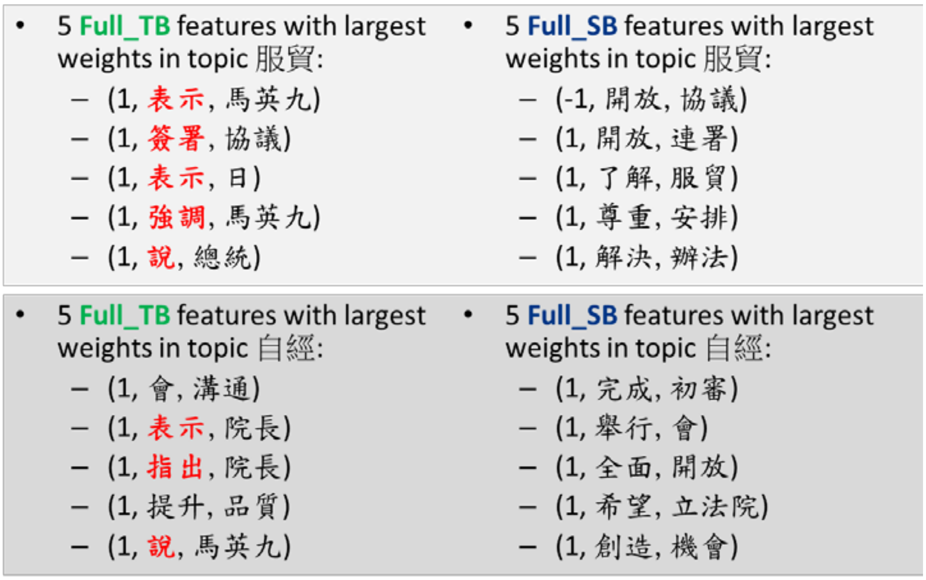
### 5.2.1 Performance of type of feature and merged feature



| | MajorClass | BOW | BiWord | TriWord | PT_SB | Full_SB | PT_TB | Full_TB | Merge |
|---|---|---|---|---|---|---|---|---|---|
| 服貿 | 0.532 | 0.675 | 0.759 | 0.725 | 0.623 | 0.612 | 0.673 | 0.713 | 0.769 |
| 自經 | 0.534 | 0.786 | 0.815 | 0.706 | 0.706 | 0.743 | 0.791 | 0.769 | 0.838 |
| 美牛 | 0.625 | 0.727 | 0.797 | 0.757 | 0.642 | 0.696 | 0.675 | 0.750 | 0.831 |
| 基薪 | 0.649 | 0.681 | 0.737 | 0.770 | 0.576 | 0.611 | 0.672 | 0.717 | 0.765 |

Figure 6. Performance of each type of feature and merged feature.

From Figure 6, we can find that word-based features(red) are still the most discriminative feature for stance classification, and Bi-Word feature even has the best performance in 3 of 4 topics. Besides, dependency features filtered by part-of-speech tags (PT_TB and Full_TB, in green color) outperform the dependency features filtered by sentiment lexicon (PT_SB and Full_SB, in blue color). In order to clarify the reason, we printed out five Full_TB and Full_SB features with largest weights in topic 服貿 and 自經 in Figure 7, where the larger weight means the higher tendency to agree the stance statement.



- 5 Full_TB features with largest weights in topic 服貿:
  - (1, 表示, 馬英九)
  - (1, 簽署, 協議)
  - (1, 表示, 日)
  - (1, 強調, 馬英九)
  - (1, 說, 總統)
- 5 Full_SB features with largest weights in topic 服貿:
  - (-1, 開放, 協議)
  - (1, 開放, 連署)
  - (1, 了解, 服貿)
  - (1, 尊重, 安排)
  - (1, 解決, 辦法)
- 5 Full_TB features with largest weights in topic 自經:
  - (1, 會, 溝通)
  - (1, 表示, 院長)
  - (1, 指出, 院長)
  - (1, 提升, 品質)
  - (1, 說, 馬英九)
- 5 Full_SB features with largest weights in topic 自經:
  - (1, 完成, 初審)
  - (1, 舉行, 會)
  - (1, 全面, 開放)
  - (1, 希望, 立法院)
  - (1, 創造, 機會)

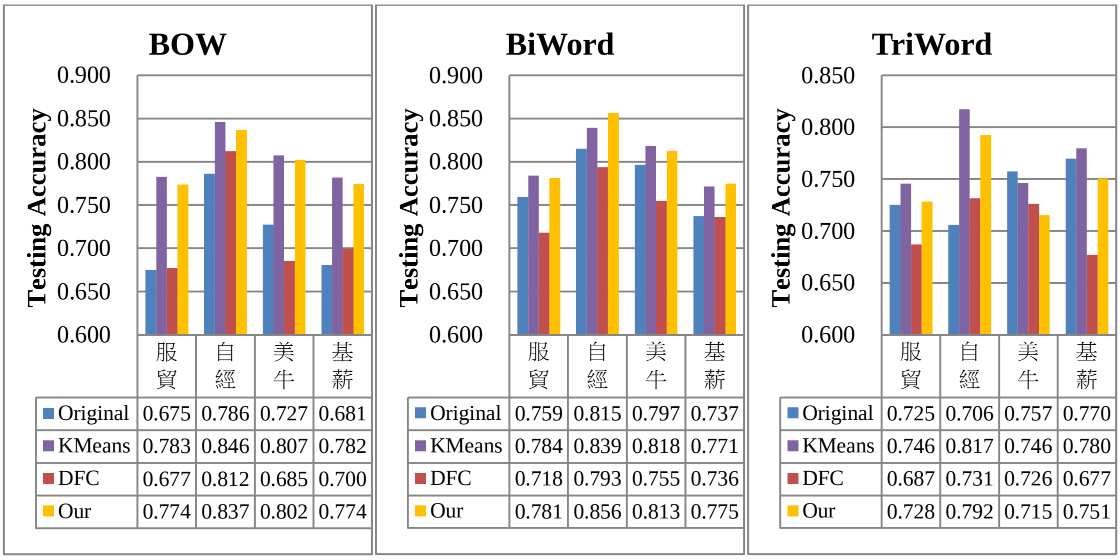
Figure 7. Top 5 features with largest coefficient (weight) in linear classifier for Full_TB(left) and Full_SB(right) feauture and topic 服貿(top) and 自經(buttom).

The red word in Figure 7 is the head word not in sentiment lexicon (i.e. the neutral word), such as "表示", "簽署". On left-hand side we show 5 features with largest weights of Full_TB features and on right-hand side we show the corresponding ones of Full_SB features. Firstly, some of red words belong to "opinion operator" category [18], such as "表示", "強調", "說" and "指出", which mean actions to express opinions. Although those words are neutral, the combination with other words (especially with important person) in dependency relation has discriminative power for classifying stance in news domain. This phenomenon implies authors of news tend to use neutral words when considering ethics of journalism. However, they may still have their own stance so they expressed it implicitly. News is less subjective, so stance classification becomes more challenging in news domain.

Since part-of-speech based dependency features yielded better results, we merged BOW, Bi-Word, Tri-Word, Full_TB and PT_TB as our final features. In 3 of 4 topics, performance of merged features (yellow) increased.

### 5.2.2 Performance of direct feature clustering and our proposed approach

The first step of our proposed approach is to divide features into two sides, one side is for the features whose coefficient is larger than or equal to zero and another side is for the features whose weight is less than zero. This is a crucial step if we are merging similar texts for further classification. In this section, we will show the performance of direct feature clustering and our proposed approach, and we will do real cases that discriminative features are erased if we ignore first step.



BOW

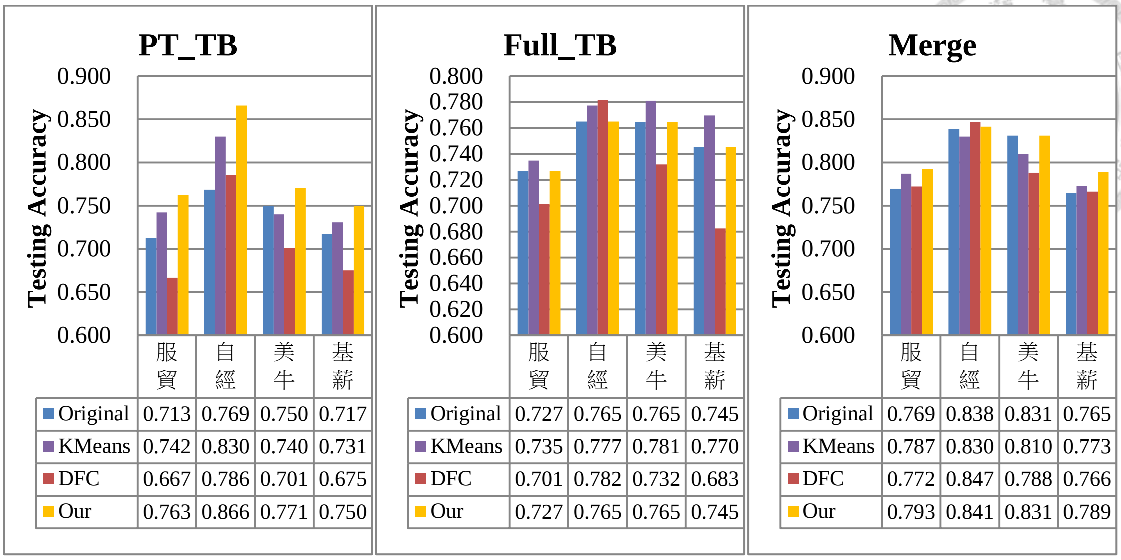
| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| 🟦 Original | 0.675 | 0.786 | 0.727 | 0.681 |
| 🟪 KMeans | 0.783 | 0.846 | 0.807 | 0.782 |
| 🟥 DFC | 0.677 | 0.812 | 0.685 | 0.700 |
| 🟨 Our | 0.774 | 0.837 | 0.802 | 0.774 |

BiWord

| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| 🟦 Original | 0.759 | 0.815 | 0.797 | 0.737 |
| 🟪 KMeans | 0.784 | 0.839 | 0.818 | 0.771 |
| 🟥 DFC | 0.718 | 0.793 | 0.755 | 0.736 |
| 🟨 Our | 0.781 | 0.856 | 0.813 | 0.775 |

TriWord

| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| 🟦 Original | 0.725 | 0.706 | 0.757 | 0.770 |
| 🟪 KMeans | 0.746 | 0.817 | 0.746 | 0.780 |
| 🟥 DFC | 0.687 | 0.731 | 0.726 | 0.677 |
| 🟨 Our | 0.728 | 0.792 | 0.715 | 0.751 |



| PT_TB | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| 🟦 Original | 0.713 | 0.769 | 0.750 | 0.717 |
| 🟪 KMeans | 0.742 | 0.830 | 0.740 | 0.731 |
| 🟥 DFC | 0.667 | 0.786 | 0.701 | 0.675 |
| 🟨 Our | 0.763 | 0.866 | 0.771 | 0.750 |

| Full_TB | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| 🟦 Original | 0.727 | 0.765 | 0.765 | 0.745 |
| 🟪 KMeans | 0.735 | 0.777 | 0.781 | 0.770 |
| 🟥 DFC | 0.701 | 0.782 | 0.732 | 0.683 |
| 🟨 Our | 0.727 | 0.765 | 0.765 | 0.745 |

| Merge | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| 🟦 Original | 0.769 | 0.838 | 0.831 | 0.765 |
| 🟪 KMeans | 0.787 | 0.830 | 0.810 | 0.773 |
| 🟥 DFC | 0.772 | 0.847 | 0.788 | 0.766 |
| 🟨 Our | 0.793 | 0.841 | 0.831 | 0.789 |

Figure 8. Performance of original raw feature (blue), direct feature clustering using K-Means (purple), direct feature clustering using community detection (red) and our approach (yellow)

The performance of direct feature clustering and our proposed approach are shown in Figure 8. We can firstly observe the performance of original raw feature (blue), direct feature clustering using K-Means (red) and our proposed solution. Although K-Means is a very effective solution for clustering features, it is much slower than Lourvain community detection algorithm. The average time complexity of K-Means in Scikit-learn's implementation is O(KnTd) where K is the number of centroids, n is number of samples, T is maximum iteration, and d is the dimension of vector. On the other hand, the time complexity in the worst case of Lourvain's method is O(E), where E is the number of edges in graph. The real computation of Lourvain's method is usually much less than O(E) because calculating difference of modularity is almost constant, and once a node is absorbed by another node, all adjacent edges don't need to be evaluated.

In our framework, we have to do feature clustering for each training phrase, including every training phrase while searching best parameters using cross-validation. It is almost impossible to finish tasks if we choose K-Means algorithm; as a consequence, a more efficient clustering algorithm, such as Lourvain community detection algorithm, was applied in our framework.

After deciding to use Lourvain community detection algorithm, we can observe the results of original raw features (blue), direct feature clustering using community detection (red), and our proposed solution. Experimental results show that in most of cases our approach outperforms the other twos. Moreover, in approximate half of cases, direct feature clustering lead to worse performance than original raw features, such as PT_TB feature in topic 服貿, 美牛, 基薪 and *BiWord* feature in all topics. It shows that dividing features into two sides using the weights of linear classifier is a crucial step. In order to investigate how performance decreases if we cluster features in two sides into same cluster, we plot the accuracy decrease (comparing to raw feature) of *Full_TB* feature in topic 服貿 for each edge threshold in Figure 9.

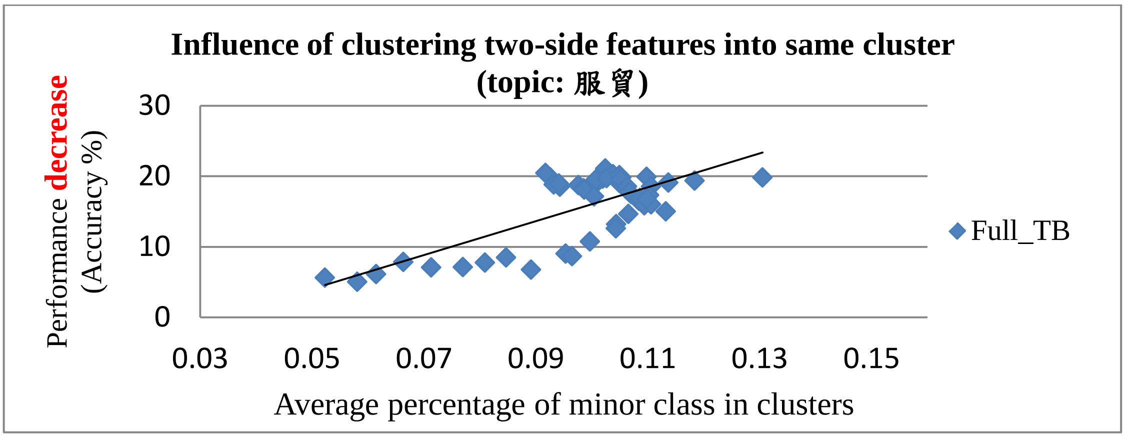
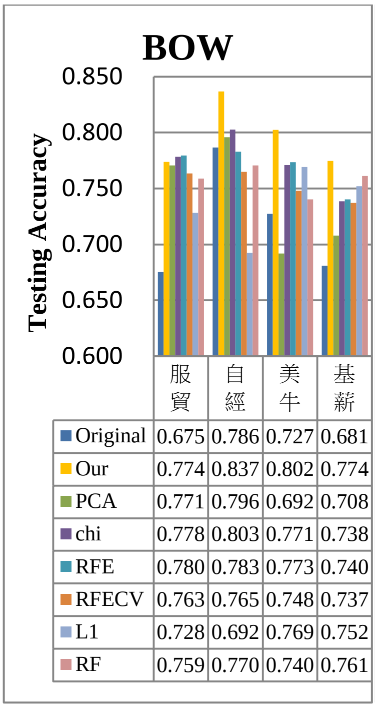


Figure 9. Performance decreasing amount when cluster two-side features into same cluster.

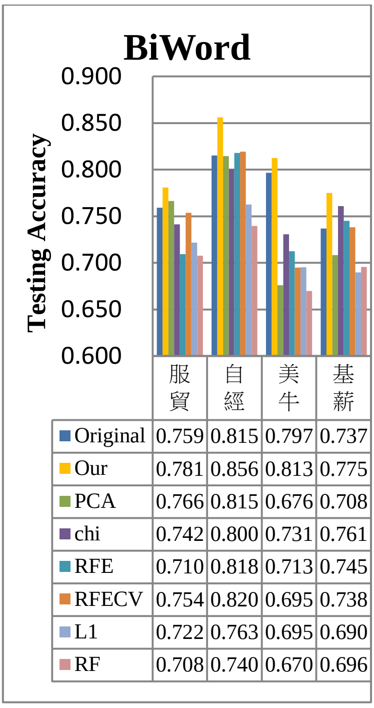
In Figure 9, y-axis denotes the decreasing amount(%) of accuracy comparing to raw features, and smaller value means better accuracy. X-axis denotes the average percentage of minor class (w>=0 or w<0) in clusters, and larger value means more features from two sides are clustered into same cluster. Besides, each point denotes one result of certain edge threshold. The trend shows that when more features from two sides are clustered into same cluster, the worse performance the feature clustering is. This result confirms the importance of the first step in our approach.

### 5.2.3 Performance of baseline approaches and our proposed approach

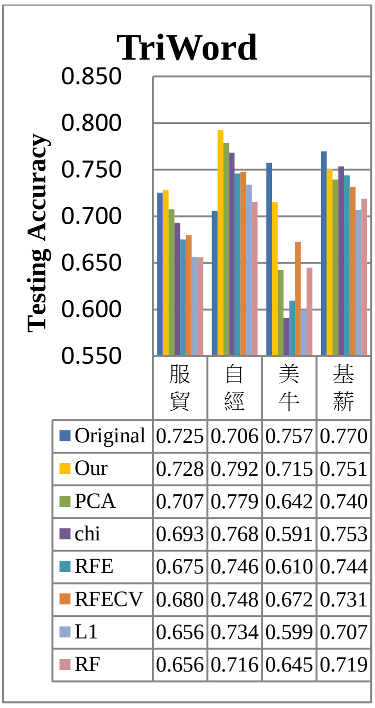
After comparing our approach to direct feature clustering, we compare the performance of our approach to other baseline approaches.



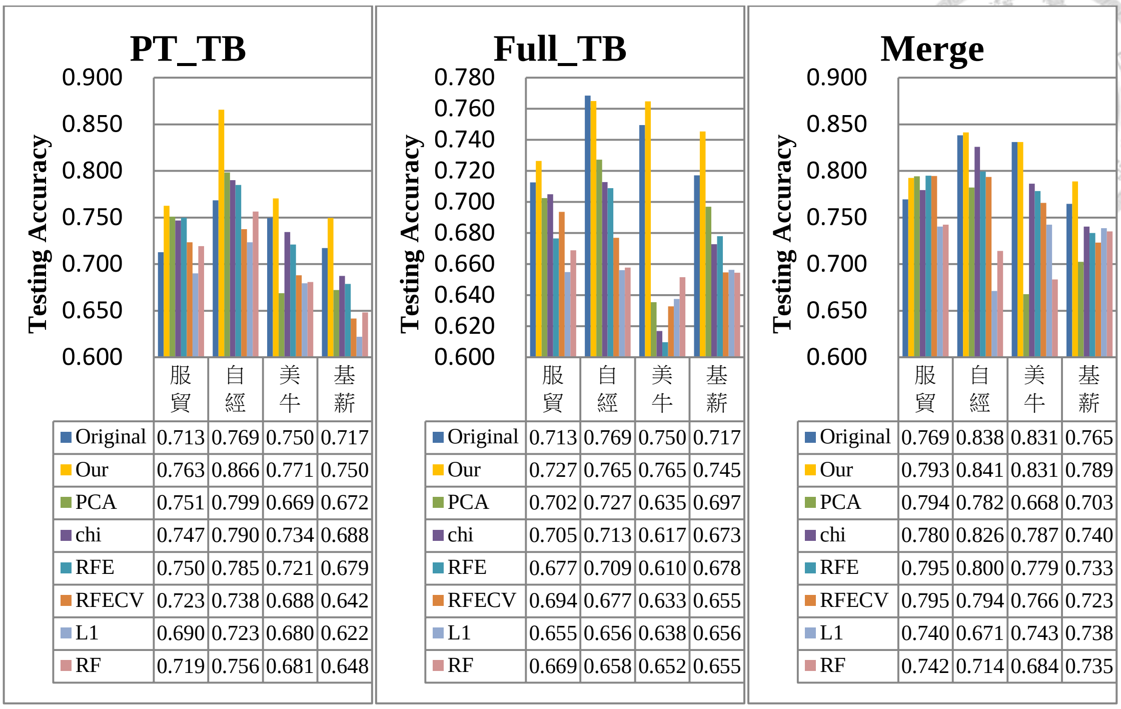
| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| Original | 0.675 | 0.786 | 0.727 | 0.681 |
| Our | 0.774 | 0.837 | 0.802 | 0.774 |
| PCA | 0.771 | 0.796 | 0.692 | 0.708 |
| chi | 0.778 | 0.803 | 0.771 | 0.738 |
| RFE | 0.780 | 0.783 | 0.773 | 0.740 |
| RFECV | 0.763 | 0.765 | 0.748 | 0.737 |
| L1 | 0.728 | 0.692 | 0.769 | 0.752 |
| RF | 0.759 | 0.770 | 0.740 | 0.761 |



| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| Original | 0.759 | 0.815 | 0.797 | 0.737 |
| Our | 0.781 | 0.856 | 0.813 | 0.775 |
| PCA | 0.766 | 0.815 | 0.676 | 0.708 |
| chi | 0.742 | 0.800 | 0.731 | 0.761 |
| RFE | 0.710 | 0.818 | 0.713 | 0.745 |
| RFECV | 0.754 | 0.820 | 0.695 | 0.738 |
| L1 | 0.722 | 0.763 | 0.695 | 0.690 |
| RF | 0.708 | 0.740 | 0.670 | 0.696 |



| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| Original | 0.725 | 0.706 | 0.757 | 0.770 |
| Our | 0.728 | 0.792 | 0.715 | 0.751 |
| PCA | 0.707 | 0.779 | 0.642 | 0.740 |
| chi | 0.693 | 0.768 | 0.591 | 0.753 |
| RFE | 0.675 | 0.746 | 0.610 | 0.744 |
| RFECV | 0.680 | 0.748 | 0.672 | 0.731 |
| L1 | 0.656 | 0.734 | 0.599 | 0.707 |
| RF | 0.656 | 0.716 | 0.645 | 0.719 |



PT_TB

| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| ■ Original | 0.713 | 0.769 | 0.750 | 0.717 |
| ■ Our | 0.763 | 0.866 | 0.771 | 0.750 |
| ■ PCA | 0.751 | 0.799 | 0.669 | 0.672 |
| ■ chi | 0.747 | 0.790 | 0.734 | 0.688 |
| ■ RFE | 0.750 | 0.785 | 0.721 | 0.679 |
| ■ RFECV | 0.723 | 0.738 | 0.688 | 0.642 |
| ■ L1 | 0.690 | 0.723 | 0.680 | 0.622 |
| ■ RF | 0.719 | 0.756 | 0.681 | 0.648 |

Full_TB

| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| ■ Original | 0.713 | 0.769 | 0.750 | 0.717 |
| ■ Our | 0.727 | 0.765 | 0.765 | 0.745 |
| ■ PCA | 0.702 | 0.727 | 0.635 | 0.697 |
| ■ chi | 0.705 | 0.713 | 0.617 | 0.673 |
| ■ RFE | 0.677 | 0.709 | 0.610 | 0.678 |
| ■ RFECV | 0.694 | 0.677 | 0.633 | 0.655 |
| ■ L1 | 0.655 | 0.656 | 0.638 | 0.656 |
| ■ RF | 0.669 | 0.658 | 0.652 | 0.655 |

Merge

| | 服貿 | 自經 | 美牛 | 基薪 |
|---|---|---|---|---|
| ■ Original | 0.769 | 0.838 | 0.831 | 0.765 |
| ■ Our | 0.793 | 0.841 | 0.831 | 0.789 |
| ■ PCA | 0.794 | 0.782 | 0.668 | 0.703 |
| ■ chi | 0.780 | 0.826 | 0.787 | 0.740 |
| ■ RFE | 0.795 | 0.800 | 0.779 | 0.733 |
| ■ RFECV | 0.795 | 0.794 | 0.766 | 0.723 |
| ■ L1 | 0.740 | 0.671 | 0.743 | 0.738 |
| ■ RF | 0.742 | 0.714 | 0.684 | 0.735 |

Figure 10 . Performance of baseline approaches and our proposed approach

The experimental results of baseline approaches and our proposed approach are shown in Figure 10. In most of cases of single feature, our proposed solution (yellow) outperforms original raw feature and all other baseline approaches. Our solution even increase accuracy up to near 10% in PT_TB feature and topic 自經. Although in merged feature our approach cannot outperform baseline approaches in topic 服貿 and 美牛, the performance is almost the same to those baselines. Besides, in other two topics our approach can still outperform all other baselines.

In summary, our approach enhance the performance of stance classification in most of cases, and it avoid the problem of mixing features with different stance tendency.

## 5.3 Result Analysis

### 5.3.1 Sensitivity of the threshold while building feature-to-feature graph

Similarity threshold while building feature-to-feature graph is the only parameter in our solution, so we investigate how the testing accuracy change over different threshold. The results are shown in Figure 11 and Figure 12.

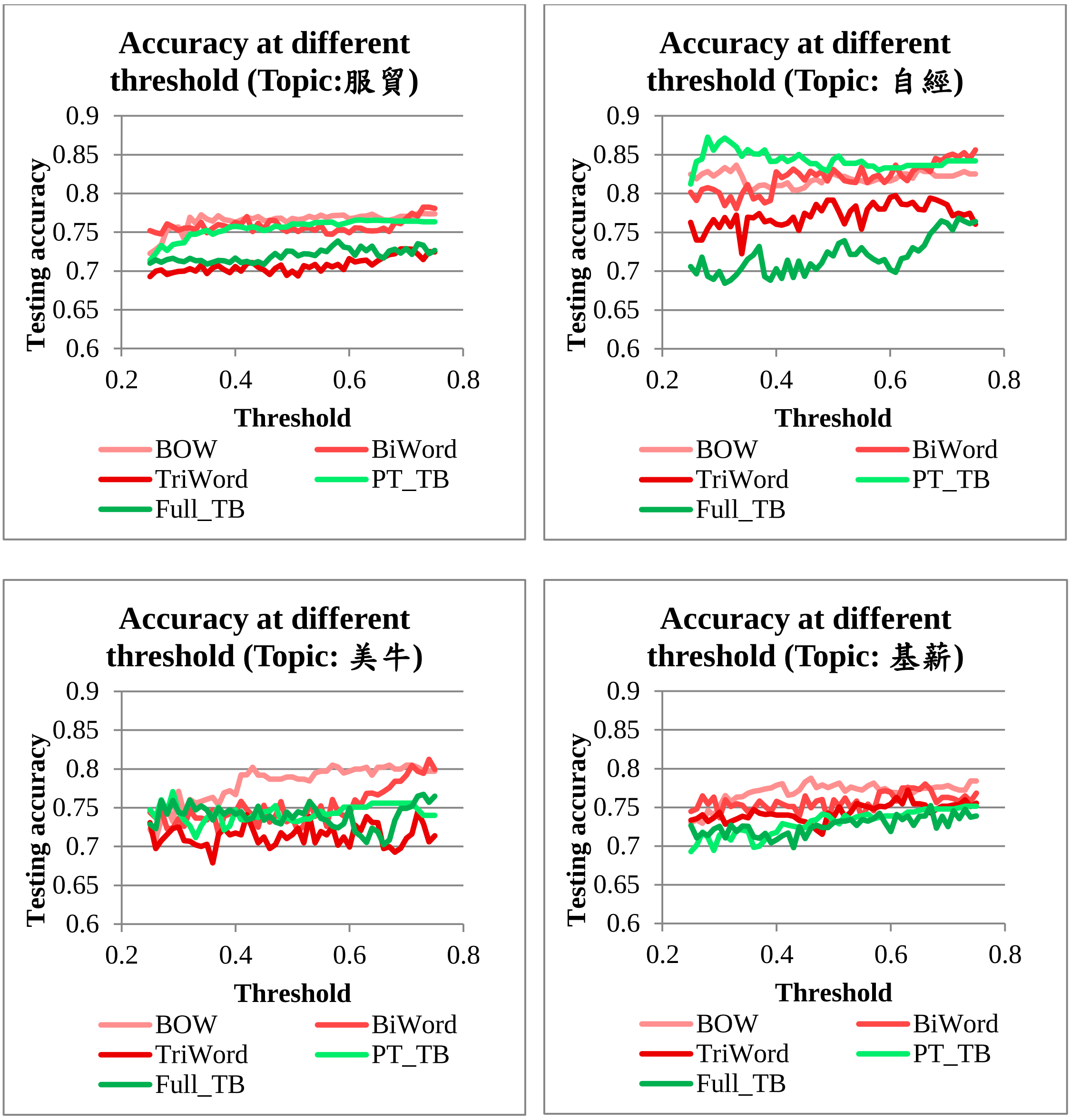


Figure 11. Testing accuracy at different threshold for each topic and single feature.

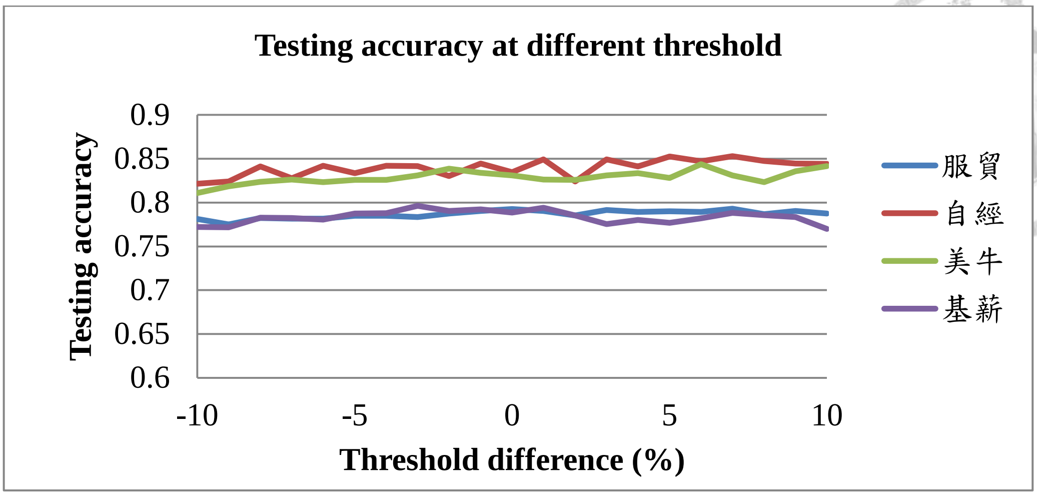


Figure 12. Testing accuracy at different threshold difference for merged features.

In Figure 11, we plotted the testing accuracy at different threshold for each topic and each feature. In general, higher threshold can produce better accuracy, but there is no clear trend to indicate how to select best threshold. As a consequence, we used cross-validation to choose best threshold. Besides, we can observe that the accuracy changed more largely for the topic which has fewer documents, not only because the denominator is smaller but also because it is more difficult to train a stable classifier when training data is pretty small. For example, we only have 116 documents in topic 自經, which is not enough for this difficult problem.

In Figure 12, we also plotted the testing accuracy at different threshold difference for merged feature, and results show that there is no significant trend telling us how to select best threshold.

# Chapter 6  Conclusion and Future Work

In our research, we collected news articles of controversial topics from Chinese online news websites, and we built an annotation website for people to annotate the stance of selected news articles. We provided a new dataset for future research of stance classification on Chinese news domain. In our experiments, we found that word-based features are still very important for classifying stance. Besides, neutral words in dependency features play key roles for stance classification in news domain but these features were usually ignored in previous sentiment prediction works. Finally, we propose a simple and efficient approach to merge text features by incorporating information of unlabeled data, and results show that our approach outperforms other raw features and other baseline approaches in most of cases. In addition, our approach can avoid the problem that merging features without considering stance tendency may leads to worse performance.

There are several important directions to improve our work. In step 1, we can try soft constraints when dividing features into groups, which mean that we may not have to remove all edges between groups. Besides, in step 2, the addition of vectors may have word ordering problem, which means (A, B) and (B, A) feature has exact same auxiliary vector but it is unreasonable. In step 3, the best threshold of each different group of feature in our approach is different, so we can improve our solution if we find a way to decide best threshold for each type of feature. Finally, K-Means can produce better clustering results but it is not efficient enough in our framework, so we should find the clustering algorithm which is as effective as K-Means and as efficient as Lourvain's community detection algorithm to produce best results.

# REFERENCE

[1] Hasan, Kazi Saidul and Ng, Vincent. "Why are You Taking this Stance? Identifying and Classifying Reasons in Ideological Debates," EMNLP, 2014.

[2] Somasundaran, Swapna and Wiebe, Janyce. "Recognizing Stances in Online Debates," ACL/IJCNLP, 2009.

[3] Swapna Somasundaran and Janyce Wiebe. 2010. "Recognizing stances in ideological on-line debates," NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text (CAAGET '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 116-124.

[4] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael Minor. 2011. "Cats rule and dogs drool!: classifying stance in online debate," 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA '11). Association for Computational Linguistics, Stroudsburg, PA, USA, 1-9.

[5] Walker, Marilyn A., Anand, Pranav, Abbott, Rob and Grant, Ricky. "Stance Classification using Dialogic Properties of Persuasion," HLT-NAACL, 2012.

[6] Sridhar, Dhanya and Getoor, Lise and Walker, Marilyn. "Collective Stance Classification of Posts in Online Debate Forums". ACL Joint Workshop on Social Dynamics and Personal Attributes in Social Media, 2014.

[7] Kazi Saidul Hasan and Vincent Ng. "Extra-Linguistic Constraints on Stance Recognition in Ideological Debates," the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 816-821, 2013.

[8] Hasan, Kazi Saidul and Ng, Vincent. "Frame Semantics for Stance Classification,"

CoNLL, 2013.

[9] Thomas, Matt, Pang, Bo and Lee, Lillian. "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts," EMNLP, 2006

[10] A. Yessenalina, Y. Yue, and C. Cardie, "Multi-level Structured Models for Document-level Sentiment Classification", EMNLP, 2010

[11] Balahur, Alexandra, Kozareva, Zornitsa and Montoyo, Andrés. "Determining the Polarity and Source of Opinions Expressed in Political Debates," CICLing, 2009.

[12] Murakami, Akiko and Raymond, Rudy. "Support or Oppose? Classifying Positions in Online Debates from Reply Activities and Opinion Expressions," COLING (Posters), 2010.

[13] Amita Misra, Marilyn A. Walker "Topic Independent Identification of Agreement and Disagreement in Social Media Dialogue," SIGDIAL, 2013

[14] Qiu, Minghui, Yang, Liu and Jiang, Jing. "Modeling interaction features for debate side clustering," CIKM, 2013.

[15] Krippendorff, K. (2011). "Computing Krippendorff's Alpha-Reliability," Retrieved from http://repository.upenn.edu/asc_papers/43

[16] Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. “Discriminative Reordering with Chinese Grammatical Relations Features,” Third Workshop on Syntax and Structure in Statistical Translation.

[17] Ku, Lun-Wei, Lee, Li-Ying and Chen, Hsin-Hsi. "Opinion extraction, summarization and tracking in news and blog corpora," AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs, 2006.

[18] Ku, Lun-Wei, Wu , Tung-Ho, Lee, Li-Ying and Chen, Hsin-Hsi. "Construction of an Evaluation Corpus for Opinion Extraction," NTCIR-5 Workshop Meeting, Tokyo, Japan, 2005.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation

of Word Representations in Vector Space," In Proceedings of Workshop at ICLR, 2013.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and their Compositionality," In Proceedings of NIPS, 2013.

[21] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations," In Proceedings of NAACL HLT, 2013.