

國立政治大學資訊科學系

Department of Computer Science

National Chengchi University

碩士論文

Master's Thesis

深度學習對於中文句子的表示  
Sentence Representation in Chinese

研究生：管芸辰

指導教授：蔡銘峰

中華民國一百零六年十一月

November 2017




深度學習對於中文句子的表示  
Sentence Representation in Chinese

研究生：管芸辰      Student：Yun Chen Kuan  
指導教授：蔡銘峰      Advisor：Ming-Feng Tsai

國立政治大學

資訊科學系

碩士論文

The seal of National Chengchi University is a large, faint watermark in the background. It is a circular emblem with the university's name in Chinese characters '國立政治大學' around the top and 'National Chengchi University' around the bottom. In the center is a stylized cloud or flower-like shape.

A Thesis  
submitted to Department of Computer Science  
▪ National Chengchi University ▪  
in partial fulfillment of the Requirements  
for the degree of  
Master  
in  
Computer Science

中華民國 一百零六 年 十一 月  
November 2017

## Abstract

The paper demonstrate the popular method in recent years to construct the semantic embedding, and use classification to verify the accuracy of these models on Chinese.



# Content

<b>Abstract</b>	<b>1</b>
1 Introduction . . . . .	1
1.1 Abstract . . . . .	1
1.2 Purpose . . . . .	1
2 Related Work . . . . .	2
3 Methods . . . . .	3
3.1 The model introduction . . . . .	4
4 Conclusion . . . . .	5
4.1 Experiment Settings . . . . .	5
5 Discussion . . . . .	7
5.1 Discussion . . . . .	7

# Figure Content



# Table Content

Table 3.1	Tag Category . . . . .	3
Table 4.1	Results . . . . .	5
Table 4.2	FastText . . . . .	6
Table 4.3	ResultsDoc2Vec . . . . .	6



# Chapter 1

## Introduction

### 1.1 Abstract

How to make the sentence embedding with its own semantics more precisely is study of interest, since it's beneficial for several NLP tasks like machine translation, sentiment analysis. Since the internet text volume grows so enormously and rapidly, how to make the information can be extracted more efficiently and precisely become more critical for many applications. Chinese forums, blogs or microblog expand especially rapidly. The studies tried to vectorize the sentences with deep learning approach with more general way to make it invariant to the languages properties.

Recently word2vec[8] is considered to work for evaluating word semantics in general cases. Additionally, the character is invariant to the language. Nevertheless, the embedding in sentence level is more complicated, it's related to the sentence structure, intention or context. There are several methods raised in recent years, like Siamese-CBOW, FastText ...etc. Most of them are able to train batch of text to construct semantic vectors.

### 1.2 Purpose

So far, most the studies are conducted in English or more general way to applied in various languages ,since The most platforms are contributed by the worldwide users. Most approaches also are aimed at being invariant to language properties. However, few of them evaluate the effectiveness of these approaches when it comes to Chinese. we are also interested if the feature also works in Chinese or other languages, and if the algorithm is invariant to the language grammar.



# Chapter 2

## Related Work

In recent years, most models are aimed at English or more general way. [10] performed the basic way to classify the articles from WeiBo with Naive Bayes.

Additionally, sentiment analysis with typical deep learning models are conducted, like CNN [6], RNN [1].

When it comes to multilingual environment, the preprocess approach may differ in languages. Like Chinese and Japanese, segmentation may also involved. In the example of FastText[4], they also demonstrated to convert character into pinyin, which make the subword information can be obtained.

There is also a work[9] to evaluate the multilingual approach and monolingual one. However, it used the Spanish and English as target, both two are belongs to Indo-European languages. It also addressed the culture difference, "dragon" mean harmful in English but it's opposite in Chinese.

# Chapter 3

## Methods

The data set we chose is Open WeiboScope, which is collected WeiBo randomly by researchers at the Journalism and Media Center of the University of Hong Kong in 2012. It contains 226 millions posts distributing over the year. We used the tags in post as the indicators of sentiment, and removed some duplicated posts or some posts without any tags, or too many tags. We evaluated the accuracy of the classification for different algorithms. We used the TF-IDF and SVM (Joachims, 1998). as baseline.

For the data preprocessing and cleansing, it's a Weibo feature to allow the user to use emoticon, and the emoticon in raw data expressed as [笑](smile),[淚](tear).

we removed the posts that contains too many tags, or without any tags. We also removed the duplicated posts by their post id roughly because it is a property of Chinese microblog [3] for Chinese netizens to post repeatedly, but most algorithms can't resist the duplicated posts. Besides, we only chose the post that over certain length .

The posts meets the criteria is about 7.4 millions. And we removed the tags in the original post, and there are so many tags , we use most-used 6 categories to categorize them as below.

Table 3.1: Tag Category

JOY	呵呵 酷 贊 鼓掌 耶
DISGUST	黑 汗
SAD	可憐 淚 衰 失望 心 生病 囧 鄙 淚 衰 失望 心 生病 囧 鄙
FEAR	委屈 可憐
SURPRISE	吃驚 吃驚
ANGER	怒 抓狂

We used jieba and dictionary to segment to post.

### 3.1 The model introduction

#### 1. TF-IDF + SVM

The conventional way to evaluate the semantics based on the occurrence of words and term, and it also takes the occurrence of word in global context into consideration. It's simple and effective, but it still suffers from some disadvantages like data sparsity and high dimension.

#### 2. FastText [4]

The structure of FastText is similar to CBOW of Mikolov et al. (2013), and it uses the softmax to compute the probabilities for predefined classes. The word representation is looked up through a table and finally averaged into the text representation. Finally it uses the linear classification.

#### 3. Paragraph vector [7]

This is method is purposed in [7]. The idea is obtain the summary of paragraphs, sentences or documents. There are 2 different algorithms we tested, which are dm (distributed memory) and dbow(distributed bag of words).

We use the implementation of Gensim and use SVM with linear kernel to classify.

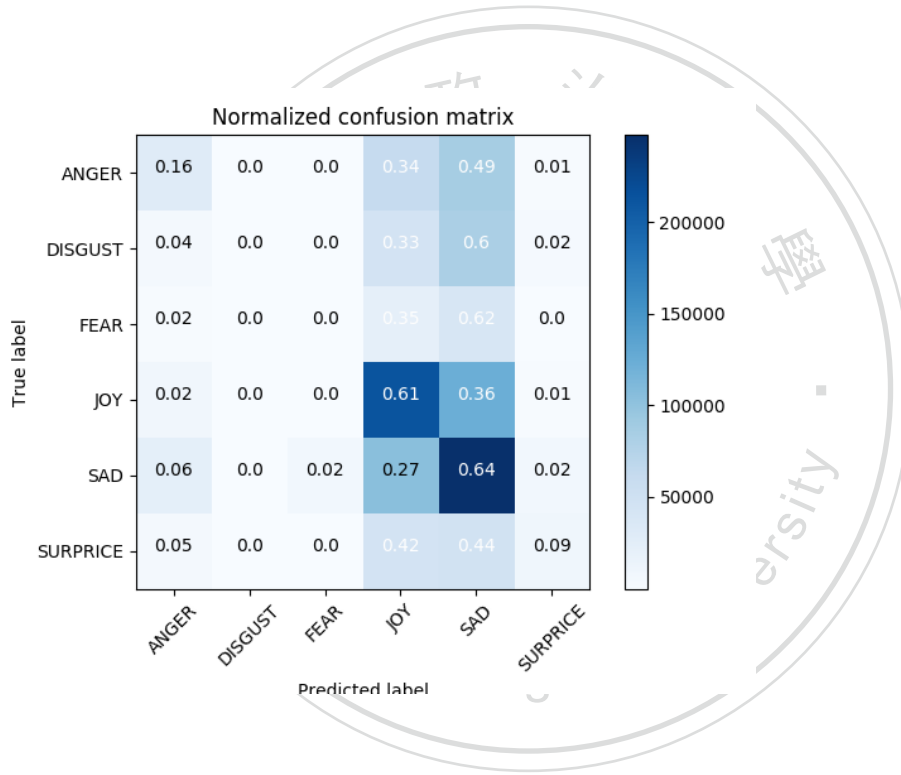
#### 4. Siamese-Cbow[5]

The method computes a sentence embedding is to average the embeddings of its constituent words, instead of using pre-trained word embedding.

We used the implementation (<https://bitbucket.org/TomKenter/siamese-cbow/overview>) from the author, and made it compatible with python3 for better compatibility with unicode.

# Chapter 4

## Conclusion



### 4.1 Experiment Settings

We used baseline TD-IDF plus SVM with linear kernel as baseline. We compared it with other models with different settings.

Table 4.1: Results

Tf-IDF	$0.44 \pm 0.04$
PVDM(dbow)	0.40
FastText	0.51
FastText(Pinyin)	0.51
Simaese-CBOW	$0.41 (\pm 0.04)$

Table 4.2: FastText

	8	12	16	32	64
no segmentation	0.369	0.375	0.389	0.372	0.368
segmentation	0.515	0.515	0.514	0.516	0.513
segmentation + pinyin	0.513	0.518	0.516	0.517	0.51

Table 4.3: ResultsDoc2Vec

	Test set	Training Set
dm/c	0.384	0.384
dbow	0.404	0.457
dm/m	0.38	0.436

For PVDB, we use 3 different models dm/c and dm/m and dbow. All of them, we choose most commonly-used parameters, dm : dimension:100, window size:10, negative:5, hs:0 and we tested both dm with concatenation of context vectors (dm/c) and average of context vectors(dm/m). The other model dbow, we chose the same parameters.

In FastText experiment, we iterated through the parameters like window size from 8 to 100, loss function ns,hs,softmax. Since the result didn't indicate significant difference between these parameters, we only display 1 of them as reference.

Additionally, we also tried to convert data set to pinyin to evaluate if the pinyin improve the semantic recognition for FastText ,which support vocabulary expansion with subword information [2].

# Chapter 5

## Discussion

The result shows that FastText can archive better accuracy in general way.

### 5.1 Discussion

For the baseline, though TF-IDF it can archive the accuracy about  $0.44(\pm 0.04)$ . The most distinguishable features they use are some rarely used terminology. Since we only removed the duplicated post roughly, it may still suffer from the duplicated post from different sources with certain rarely-used words. In general, the model is not general enough, it may not be applicable when the data set changed.

Generally, FastText can get the better accuracy, even converting the posts to pinyin, it also achieves the same accuracy. Though, we tried the different settings for FastText, the accuracy didn't differ so much. In the comparison set, segmented data set outperforms the one without segmentation. It suggested that the term itself is more meaningful than a single character. And it also took much less time than other algorithms to complete.

The Siamese-CBOW, the performance is below the baseline. We tried evaluate the model it trained, it seemed it is not converged enough. The word embedding is not converted correctly. In the original paper, the dataset they used is Toronto Books, which contains novels, therefore the semantics of the sentences may be more coherent with previous sentence and next one.

We demonstrated the various modern methods on the Chinese corpus, and it indicated that some models like FastText are invariant to language property. In general, most models improve the semantic analysis compared with traditional TFIDF.

Most methods are developed with English property, so segmentation plays a cru-

cial role to make the Chinese posts look like English. But the segmentation may also contribute something wrong. Though FastText also can performed with non-segmented sentences, it performed worse due to the nature of word embedding.



# Reference

- [1] G. Arevian. Recurrent neural networks for robust real-world text classification. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 326–329. IEEE Computer Society, 2007.
- [2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
- [3] K.-w. Fu and M. Chau. Reality check for the chinese microblog space: a random sampling approach. *PloS one*, 8(3):e58356, 2013.
- [4] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fast-text.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.
- [5] T. Kenter, A. Borisov, and M. de Rijke. Siamese cbow: Optimizing word embeddings for sentence representations. 2016.
- [6] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [7] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents icml. 2014.
- [8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. pages 3111–3119, 2013.
- [9] D. Vilares, M. Alonso Pardo, and C. Gómez-Rodríguez. Supervised sentiment analysis in multilingual environments. 53, 05 2017.
- [10] J. Zhao, L. Dong, J. Wu, and K. Xu. Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1528–1531. ACM, 2012.