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

105碩士論文

深度學習對於中文句子的表示

政治大學資訊科學系

管芸辰

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

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

Figure Content

Table Content

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, and the new derivative or new word keep growing as well. 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[11] is considered to work for evaluating word semantics in general cases. Additionally, the character is invariant to the language. Nevertheless, the embedding in sentence or pharse 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 sematic 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 or appliable to multilanguage environment. However, few of them evaluate the effectiveness of these approaches to other languages, or they evaluate the multiligual dataset without considering the characteristic of other languages. we are also interested if those models also works in

Chinese or other languages, and if the algorithm is invariant to the language grammar or language property.

Chapter 2

Related Work

2.1 Traditional Approach

[3] summarized both corpus-base and lexicon-base techniques purposed recently and listed the languages those techniques aiming at, and there are some innovative mothods combined both approaches. The advantage of corpus-base is that it is dictionary-free, but it requires relatively larger corpus to build the model, while lexicon-based approach depends mainly on existing resources to detect the sentiment. When lexicon-based approach comes to the informal articles contributed by netizens, it may suffer some troubles like misspelling, abbreviation ,words or metaphor...etc, neither can it take the sequence of words into consideration.

The basic corpus-based approach like TF-IDF is considered to be able to generate relatively good precision. However, both approaches may utilize some keywords in the sentences rather than sentiment of sentence itself. In real world, we often use negation or irony to present our feeling rather than solely keywords. To solve the problems from rapidly-evolved languages, there are both semi-supervised and unsupervised approaches introduced as well.

2.2 Chinese Related Sentiment Analysis

In recent years, most models are aimed at English or more general way.

When it comes to multiligual environment, the preprocess approach may differ in languages. The traditional ways to counter the variation of words like stemming or lemmatization are appliable to most Latin languages. Howerver, in Chinese and Japanese, segmentation may also be invloved. In the example of FastText[5], they also demostrated to convert character into pinyin, which make the subword infomation can be obtained.

Though most approaches are tested and verified by English dataset, there are some work to test in Chinese dataset as well.

[15] performed the basic way to classify the articles from WeiBo with Naive Bayes and smoothing with Laplace smooth. In this work, the authors also use emoticon as the ground truth to verify the approach, it also applied some imcrement learning.

2.3 Advanced Approach

Besides the tradition approaches, which try to extract the effective features. With deep approach, it may be possible to learn more in continuous representations. Additionally, sentiment analysis with typical deep learning models are conducted. Multiple tasks are performed like parsing (Socher et al., 2013a), language modeling (Bengio et al., 2003; Mnih and Hinton,2009) and NER (Turian et al., 2010). For sentiment analysis, the legendacy models were also tested, like CNN [7], RNN [1], but most of them are applied in English dataset only.

Recently, word2vec(Mikolov et al. 2013)[11] is considered to perform efficiently to vecctorize the meaning of single words. It is a log-bilinear model to learn continuous representations of words on very large corpora efficiently. Therefore, the same concept can be adapted to phrase or sentence level as well. Mikolov also purpose it with sentence level [9] called PVDB, and claimed it can applied to both short text and long article. These approaches are unsupervised, but it can be conducted to sentiment analysis with proper transformation.

The other way to model the sentimatic is using encoder-decoder model, which come from statistical machine translation. Skip-thoughts[8] employs the GRU encoder-decoder models. And it also combines "vocabulary expansion" from [10], they used vocabulary expansion to map word embeddings into the RNN encoder space. Therefore, it made it possible to use less vocabulary to build model, and reuse pre-trained model.

To detect the sentiment polarity of short text has attracted the interest of study as well, the most used dataset is informal tweet, which is contributed by the netizens from different background. For the task of sentiment classification, an effective feature learning method is to compose the representation of a sentence (or document) from the representations of the words or phrases it contains (Socher et al., 2013b; Yessenalina and Cardie, 2011).

Pang et al. (2002)[12] already used bag-of-word representation, and present the word as one-hot vector, and they get the better classification results. However, it is still not enough to represent the complex meaning or linguistic characteristics. The following works, like [13], the paper purpose to use deep learning to do sentiment analysis directly.

There is also a work[14] to evaluate the multiligual approach and monoligual 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

3.1 The model introduction

Here are some models we tested.

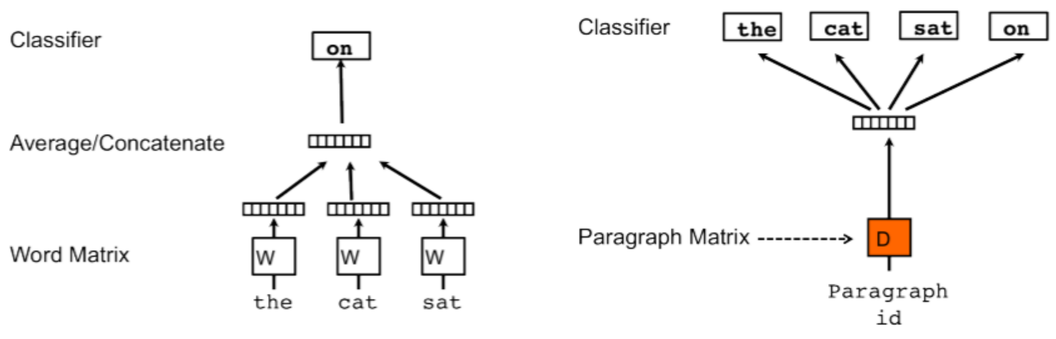
3.2 TF-IDF + SVM

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.

3.3 FastText

The approach is purposed by [5]. 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. For the words absent from word embeddings, it uses subword infomation[2] to guess the meaning of the word. Finally it uses the linear classification to classify the data.

We used the released version from Facebook github.



(a) distributed memory

(b) distributed bag of words

Figure 3.1: Paragraph vector

3.4 Paragraph Vector

This is method is purposed in [9]. 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). The DM model in figure 3.1a is quite similar with Word2Vec. Figure 3.1b show the model archeture. Compared with DM model, DBOW conceptually simple, this model requires to store less data. Two models are also can ne concatenated, which means combine both models together. The author claimed it is appliable to both short sentence and long paragraph.

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

3.5 Siamese-CBOW

The Simese-CBOW[6] computes a sentence embedding is to average the embeddings of its constituent words, instead of using pre-trained word embedding.

It applied the concept of bag of word from Word2Vec. It used the average from the word composing the sentence and use it to evaluate the possibility to predict the sentence around. The architecture shows as Figure 3.2

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.

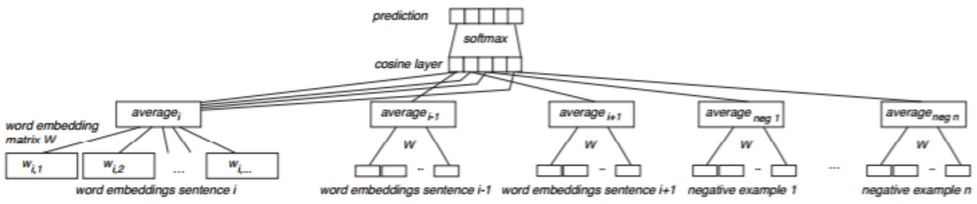


Figure 3.2: The architecture of Siamese-CBOW

Chapter 4

Experiment

4.1 Set Up

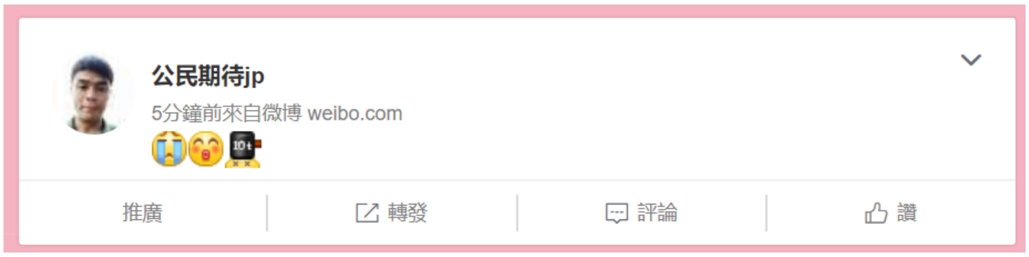


Figure 4.1: The emoticons in WeiBo

The data set we chose is Open WeiboScope[4], which is collected WeiBo randomly with API by researchers at the Journalism and Media Center of the University of Hong Kong in 2012. It contains 226 millions posts distributing evenly over the year. it's a Weibo feature to allow the user to use emoticon, and the emoticon in raw data expressed as [笑](smile),[淚](tear). It displays as images like the figure 4.1. We used the tags in posts 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.

Table 4.1: 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 吃惊
ANGER	怒 抓狂

4.2 Preprocess

For the data preprocessing and cleansing, most posts contains more than 1 tags. To avoid ambiguity, we only preserve those with single tages. we removed the posts containing 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 [4] for Chinese netizens to post repeatedly. Besides, we only chose the post that over certain length (over above 10 characters). Finally, we used jieba and dictionary to segment to post.

Like [15], we also suffered the problem that the numbers of emoticon classes skewed. The numbers of JOY and SAD are more than 50% of posts. we deleted some posts from JOY and SAD randomly to make the dataset more balance.

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 **??**.

After initial round, we found some special string or token like username or url may affect the result. Therefore, we also removed those special tokens from the posts as well.

4.3 Other

In the Paragraph vector experiment, we tested both DM and DBOW. Additionally, there are 2 different ways supported by gensim to use average or concatenation. And we used the parameters suggested.

For converting to pinyin, we use jieba + pinyin (https://www.npmjs.com/package/pinyin) npm package to counter the problem.

Chapter 5

Conclusion

5.1 Experiment Settings

We used baseline TD-IDF plus SVM with linear kernel as baseline. Since the original distribution for classes is a little skewed, most the test sample is classified into 2 major classes. We compared it with other models with different settings.

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 did'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 sematic recognition for FastText ,which support vocabulary expansion with subword information [2].

Table 5.1: Results

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

Table 5.2: 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 5.3: ResultsDoc2Vec

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

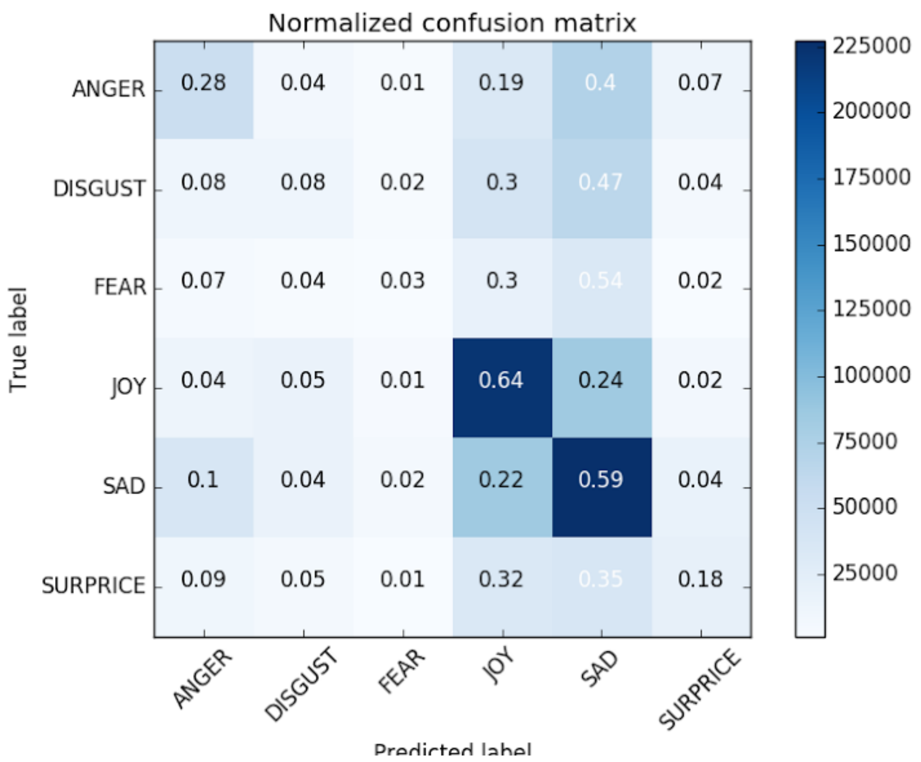


Figure 5.1: The confusion matrix for TF-IDF+ SVM

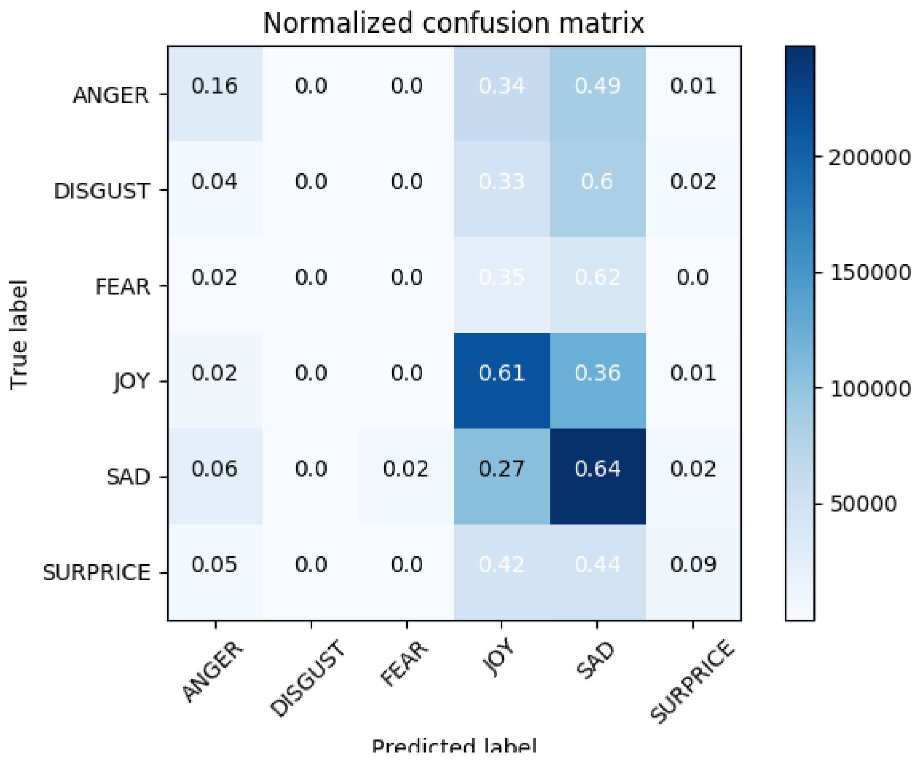


Figure 5.2: The confusion matrix for two models comparison

Chapter 6

Discussion

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

6.1 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 retrieve 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 is not different significantly despite of the various settings of loss function,window size and dimension. 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 that of other algorithms.

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. And in the confusion matrix, we found the most tested result fall into two major classes. In the original paper, the dataset they used is Toronto Books, which contains novels, therefore the sematics of the sentences may be more highly coherent with previous sentence and next one. Using some pre-trained embedding may help to deal with such situation.

Table 6.1: The feature extracted from TF-IDF

1	2	3	4	5	6
想瘦:4.40	抬杠:4.68	含:4.66	余:7.53	grunewald:4.68	印机:5.16
造謠:4.05	款:3.79	害命:4.59	符离:4.73	省:4.54	箈:4.23
浪:3.55	由此看:3.66	表同情:3.98	守:4.01	春:4.21	54:03.7
缸:3.51	淫:3.52	体:3.86	神明:3.72	前:3.83	清福:3.60
行利息:3.32	觀天象:3.42	可憐見:3.43	asce:3.56	已閱:3.35	任免:3.24
冕冠:3.24	解除:3.38	突如其:3.35	三元里:3.44	q1050505041:3.34	karei:3.07
落水狗:3.23	上刑:3.35	万念俱灰:3.11	何苦哉:3.44	伐:3.29	一:3.02
剖腹自:3.03	矢:3.32	供站:3.03	太妙了:3.34	离世:3.28	sikucd:2.97
下:3.01	超主:3.29	勞資:2.98	面三刀:3.28	噴火龍:3.27	touchsmart610:2.9
多吉:2.98	美國使館:3.06	深:2.90	slient:3.25	查無此人:3.18	翻筋斗:2.82

6.2 Baseline

We used TFIDF as baseline, the result is 0.44(±0.04), which is statistically greater than the random result in 6 classes problem, even we consider the skewed classes. We tried to evaluate the feature that TF-IDF use to classify as Table **??**.

Most of words in each classes actually are not used so commonly. It is contributed by the problem of IDF, which may overweight some rarely-used words. The other problem is that we observed the netizen keep posting something quite similar like advertisement, joke or news..., Though their content are not the same exactly, the words or terminology they used may be quite common. Besides TFIDF, most algorithms assume that the articles themselves are distinct from others. In real world, it's not a common case except some well-organized website articles.

6.3 PVDM vs. FastText

In paragraph vector experiment, the result shows that DBOW produced best accuracy among 3 models. In the original paper, the author suggested that the DM is consistently better than DBOW , and that the sum version of DM is often better than concatenation. So far, it's not clear under what condition that DBOW outperform the DM model. We tried leverge the model it built. We fetched most similar word of I (我) as Table **??**. Surprisingly, the similar words of DBOW are all not related words. Both DM/C and DM/M generated better results, which top 10 related words are synonyms of I.

Table 6.2: The most similar 5 words of I (我) in 2 models

	DM/C	DBOW	DM/M
1	俺	三	偶
2	偶	田徑運動	他
3	老子	暖人	俺
4	哀家	youtudou	我
5	皮下	化妝水	她

6.4 Subword Information

Although we converted the dataset to pinyin, the accuracy is not significantly different from the original accuracy. Intuitively, pinyin is less readible to native speaker and inreversible to the characters, since the multiple characters own the same pronunciation. Chinese contains less syllables and more homonyms than English do. In the original paper[2], they evaluate the effectiveness in various languages like Arabic, Czech, German, English...etc. All of them belong to phonography, and most languages belong to phonography. Chinese belongs to logogram specially. So far, we are still hard to quantize the effectiveness conducted in Chinese.

In the [10], it provides similar function to compensate the words absent from training set. It employs similarity of pre-trained word-embedding, rather than the similarity of n-gram characters features. There are also some differnet approaches like morphologically annotated data, whcih were introduced by Cotterell and Schütze (2015). It may be a good topic to evaluate the difference of these approaches.

6.5 Conclusion

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 sematic analysis compared with tranditional TFIDF.

Most methods are developed with English property, so segmentation plays a crucial 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.

Some publications[**?**] shows that the posters may use some morphs to avoid censorship, which we can't evaluate how these words contributes to the sentiment analysis. However, the similarity of morphs is quite easy to identified by Word2vec.

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. Dashtipour, S. Poria, A. Hussain, E. Cambria, A. Y. A. Hawalah, A. Gelbukh, and Q. Zhou. Multilingual sentiment analysis: State of the art and independent comparison of techniques. *Cognitive Computation*, 8(4):757–771, Aug 2016.

[4] K.-w. Fu and M. Chau. Reality check for the chinese microblog space: a random sampling approach. *PloS one*, 8(3):e58356, 2013.

[5] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[6] T. Kenter, A. Borisov, and M. de Rijke. Siamese cbow: Optimizing word embeddings for sentence representations. 2016.

[7] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[8] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

[9] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents icml. 2014.

[10] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

[11] 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.

[12] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[13] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.

[14] D. Vilares, M. Alonso Pardo, and C. Gómez-Rodríguez. Supervised sentiment analysis in multilingual environments. 53, 05 2017.

[15] 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.