Deep Learning Paradigm with Transformed Monolingual Word Embeddings for Multilingual Sentiment Analysis

Yujie Lu
ZhongAn Technology
Data Science Lab
luyujie@zhongan.io

Boyi Ni
ZhongAn Technology
Data Science Lab
niboyi@zhongan.io

Qijin Ji
ZhongAn Technology
Data Science Lab
jiqijin@zhongan.io

Kotaro Sakamoto
Yokohama National University
sakamoto@forest.eis.ynu.ac.jp

Hideyuki Shibuki
Yokohama National University
shib@forest.eis.ynu.ac.jp

Tatsunori Mori
Yokohama National University
mori@forest.eis.ynu.ac.jp

Abstract

The surge of social media use has triggered huge demands of multilingual sentiment analysis (MSA) for various purposes, such as unveiling cultural difference. So far, traditional methods resorted to machine translation (MT)—translating other languages to English, then adopted the existing methods of English. However, this paradigm is highly conditioned by the quality of MT. In this paper, we propose a new deep learning paradigm for MSA that assimilates the differences between languages. First, separately pre-trained monolingual word embeddings in different spaces are mapped into a shared embedding space; then, a parameter-sharing deep neural network using those mapped word embeddings for MSA is built. The experimental results justify the effectiveness of the proposed paradigm. Especially, our convolutional neural network (CNN) model with orthogonally mapped word embeddings outperforms a state-of-the-art baseline by 3.4% in terms of classification accuracy.

1 Introduction

The prevalence of social media has allowed the collection of abundant subjective multilingual texts. Twitter is such a particularly significant multilingual data source providing researchers/companies with sufficient opinion pieces on various topics/products from all over the world. By analyzing these multilingual opinion texts, there can be many useful applications, such as revealing the cultural variations and conducting customer surveys in different areas. Therefore, it's necessary to develop an effective MSA model that can process multilingual texts simultaneously.

The research on MSA has progressed slowly compared with monolingual sentiment analysis, mainly due to the lack of a benchmark dataset that can assess the cross-language adaptability of methods. As many previous studies have highlighted, open-source sentiment datasets are usually imbalanced between different languages (Mihalcea et al., 2007; Denecke, 2008; Wan, 2009; Steinberger et al., 2011). There are many freely available annotated sentiment corpora for English, but not for many other languages. As a compromise, many of the previous multilingual corpora have been built using human/machine translation, which are not authentic.

In this study, we used the MDSU corpus as our training/test dataset (Lu et al., 2017)[1]. The MDSU corpus with 5,422 tweets in total contains three languages, i.e., English, Japanese, and Chinese, and involves four identical international topics, i.e., iPhone 6, Windows 8, Vladimir Putin, and Scottish Independence. The multilinguality and topic distribution of the corpus makes it an ideal MSA dataset.

Moreover, monolingual sentiment analysis methods are usually not portable between languages,

[1]The corpus can be downloaded from https://github.com/lyjlyj517/The-MDSU-Corpus.

since they depend on language-specific polarity lexicons, POS taggers and parsers, etc. This prevents the application of many sophisticated monolingual methods to other languages, particularly the minor languages that lack basic NLP tools. This is also the reason why the most typically used methods of MSA have been based on the above-mentioned MT.

However, the MT-based paradigm is strongly conditioned by the MT quality. Considering that social media data contain many informal expressions, accurate MT is basically unguaranteed. Therefore, we proposed a new deep learning paradigm with no MT, to integrate the processing of different languages into a unified computation model. First, we pre-trained monolingual word embeddings separately; second, we mapped them into a shared embedding space; and finally, we built a parameter-sharing[2] deep neural network for MSA. A comparison between the two types of models is presented in Figure 1.

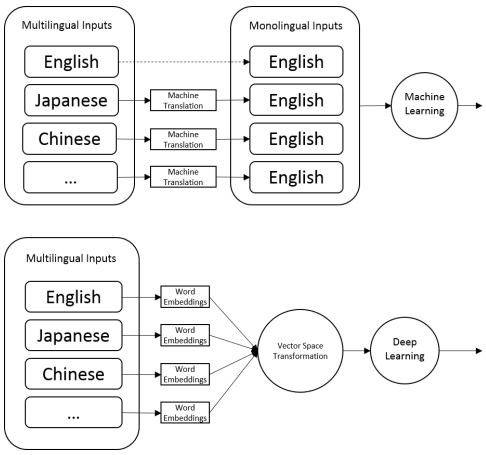

Figure 1: MT-based paradigm and the proposed deep learning paradigm

Although the study by Ruder et al. (2016) is much similar to ours in the use of deep learning methods, there are two fundamental differences. On one hand, they only input the raw monolingual word embeddings in their deep learning methods; however, we used customized pre-trained word embeddings and further transformed them into a shared space. On the other hand, they created separate models for each language, whereas we developed a single parameter-sharing model for all languages.

To the best of our knowledge, this study is the first to use a parameter-sharing deep learning paradigm with mapped word embeddings for MSA. Using such a paradigm, the only resources we required were word embeddings for each language and tokenizers for non-spaced languages (e.g., Chinese). With regard to the network structure, we mainly studied CNN in this paper.

We expected this paradigm to assimilate language differences and to make full use of the whole size of multilingual datasets. The results showed that our best parameter-sharing CNN model with orthogonally transformed word embeddings outperformed the MT-based baseline by 6.8% and a state-of-the-art baseline by 3.4%, thereby proving its effectiveness.

2 Related Work

In this section, we introduce MSA-related studies, including those on multilingual subjectivity analysis as well as the MSA of traditional text and social media.

2.1 Multilingual Subjectivity Analysis

Sentiment analysis in a multilingual framework was first conducted for subjectivity analysis. Mihalcea et al. (2007) explored the automatic generation of resources for the subjectivity analysis of a new language (i.e., Romanian). They tested a rule-based subjectivity classifier using a Romanian polarity lexicon translated from English, and Naive Bayes (NB) classifiers on a Romanian subjectivity corpus whose subjectivity is projected from its English counterpart. The results revealed that the performances of these classifiers deteriorated in Romanian compared with them in English. Banea et al. (2010) translated the English corpus into other languages (i.e., Romanian, French, English, German, and Spanish) and explored the integration of uni-gram features from multiple languages into a machine learning approach for subjectivity analysis. They demonstrated that both English and the other languages could benefit from using features from multiple languages. They believed that this was probably because, when one language did not provide sufficient information, another one could serve as a supplement.

[2]In this paper, "parameter-sharing" specifically means that the same model parameters are shared among different languages.

2.2 MSA of Traditional Text

Although there is extensive scope for improvement, translation-based methods have inspired many studies. Denecke (2008) translated German movie reviews into English, developed SentiWordNet-based methods for English movie reviews, and tested the proposed methods on the German corpus. The results revealed that the performance of the proposed methods in MSA was similar to that in monolingual settings. Wan (2009) leveraged a labeled English corpus for Chinese sentiment classification. He first machine translated the labeled English corpus and an unlabeled Chinese corpus, and then proposed a co-training approach to use the unlabeled corpus. His experimental results suggested that the co-training approach outperformed the standard inductive and transductive classifiers. Steinberger et al. (2011) annotated a valuable resource for entity-level sentiment analysis in seven European languages—English, Spanish, French, German, Czech, Italian, and Hungarian; however, their method using word polarity summation alone was preliminary and depended substantially on language-specific polarity lexicons.

2.3 MSA of Social Media

Recently, the MSA of social media content has received increasing attention. Balahur and Turchi (2013) translated English tweets into four languages—Italian, Spanish, French, and German to create an artificial multilingual corpus. They tested support vector machine (SVM) classifiers using polarity lexicon-based features on various combinations of the datasets in different languages. The results suggested that the combined use of multilingual datasets improves the performance of sentiment classification. Volkova et al. (2013) constructed a multilingual tweet dataset in English, Spanish, and Russian using Amazon Mechanical Turk. They explored the lexical variations in subjective expression and the differences in emoticon and hashtag usage by gender information in the three different languages; their results demonstrated that gender information can be used to improve the sentiment analysis performance of each language.

2.4 Comparison with Previous Work

Our study is different from the previous studies in the following ways. First, in multilingual datasets from previous studies, datasets of languages other than English have been projected from the English dataset. Banea et al. (2010) and Balahur and Turchi (2013) have used MT to obtain texts in target languages, which are considerably noisy. Mihalcea et al. (2007) and Denecke (2008) have directly used parallel corpora to eliminate this noise. However, real multilingual opinion texts would not be in the form of parallel corpora because users usually give their opinions in one language. By contrast, tweets in different languages in the MDSU corpus are real-world and covers common international topics.

As for methods, Denecke (2008) and Wan (2009) have adopted the "MT + machine learning" approach, which unavoidably imports bias during MT. The abstraction of the word feature in Balahur and Turchi (2013) can be applied to other languages, but it requires language-specific polarity lexicons. Banea et al. (2010) used uni-grams in multiple languages as features, but they might be restricted due to data sparseness issues. Volkova et al. (2013) proved the effectiveness of employing gender information, but their classifiers are not designed for multilingual settings. By contrast, our deep learning paradigm requires no polarity lexicons and can unify the representations of texts in different languages using mapped word embeddings and a deep neural network.

3 Methods

In this section, we introduce our baseline methods and the proposed deep learning method (i.e., transformed word embeddings + CNN models). The global polarity of the MDSU corpus has three types: positive, negative, and neutral; therefore, our task is technically a three-way classification problem[3].

3.1 Baselines

Our first baseline was MT-based. We used Google Translate[4] to translate Japanese/Chinese tweets into English. The SVM-based learning methods with n-

[3]For brevity, positive/negative/neutral are denoted as +/-/= respectively in Figure 2.

[4]https://cloud.google.com/translate/

gram features have been frequently used as baselines in many monolingual sentiment analysis studies (Pang et al., 2002; Go et al., 2009). Similar to their settings, we used an SVM model with a linear kernel and $C = 1$ and fed the binarized uni-gram/bi-gram term frequencies as features. The one-vs-one strategy was adopted for multiclass classification. Following the traditional paradigm, the SVM model trained on all translated tweets of the MDSU corpus is our first baseline, denoted as MT-SVM.

In addition, we re-implemented Banea et al. (2010)'s NB model that uses the cumulation of monolingual uni-gram features as our second baseline. Here, we fine tuned Banea et al. (2010)'s method in two ways: first, we used both uni-gram and bi-gram as features; and second, we used all the features instead of parts of them. We denoted this state-of-the-art baseline that does not use language-specific polarity lexicons as Banea (2010)*.

3.2 Deep Learning Paradigm

3.2.1 Word Embedding Space Transformation

Since there is no comparable open source word embeddings learned from Twitter data for multiple languages, we independently obtained word embeddings using a large number of monolingual texts for each language. However, these monolingual word embeddings were heterogeneous in terms of vector space (the meaning of each dimension was different between languages.). Hence, we attempted to reduce the discrepancies between monolingual word embeddings.

This notion was adopted from Mikolov et al. (2013). In their study, they highlighted that the same concepts have similar geometric arrangements in their respective vector spaces. This implies that if the matrix transformation is adequately performed, monolingual word embeddings in heterogeneous spaces can be adjusted to a shared vector space.

Mikolov et al. (2013) used the *Translation Matrix* (*TM*, for short) method—to obtain a linear projection between the languages using a set of pivot word pairs. Thereafter, many other ways to conduct matrix transformation have been proposed (Ruder, 2017). Following Artetxe et al. (2016), we will compare two methods: *TM* and *Orthogonal Transformation* (*OT*, for short).

Let X and Z denote the word embedding matrices for the word pairs in the bilingual dictionary, so X_i* and Z_i* are the word embeddings for the i-th entry in the dictionary. *TM* aims to identify a translation matrix $\mathbf{W}$ that minimized the following object function:

$$\underset{\mathbf{W}}{\text{minimize}} \sum_{i=1} ||X_{i*}\mathbf{W} - Z_{i*}||^2 \qquad (1)$$

On this base, *OT* further requires $\mathbf{W}$ to be an orthogonal matrix (i.e., $\mathbf{W^T W} = I$). Since both methods have analytical solutions, $\mathbf{W}$ can be efficiently computed in linear time. For *TM*, $\mathbf{W} = X^+Z$, where X^+ takes the Moore-Penrose pseudo-inverse: $(X^TX)^{-1}X^T$; for *OT*, $\mathbf{W} = VU^T$, where V and U^T can be given by the SVD fraction of Z^TX: $U\Sigma V^T$.

After $\mathbf{W}$ is identified, we map the vocabulary matrix of one language to another by multiplying it by its corespondent $\mathbf{W}$. For example, we transferred the Japanese vocabulary matrix to the English vector space using $\hat{\mathcal{X}}_{JA} = \mathcal{X}_{JA}\mathbf{W}_{JA \to EN}$ [5]. The same thing applies to Chinese.

Additionally, we also try out pre-processing before *OT* to see how it affects the performance of MSA. Specifically, we apply length normalization (*+LN*), and length normalization and mean centering together (*+LN&MC*) before *OT*.

Although these linear projections can be considered as a kind of word-level MT, this kind of space transformation is considerably less expensive than building a full-fledged MT system.

3.2.2 CNN

One of the advantages of CNNs is that they have much fewer parameters than fully connected networks with the same number of hidden units, which makes them much easier to be trained. The CNN we used is very similar to that of Kim (Kim, 2014), which is presented in Figure 2.

To unify the matrix representation of tweets in different length, the maximum length of all tweets in the dataset was used as the fixed size for tweet matrices. For shorter tweets, zero word vectors were padded at the back of a tweet matrix.

[5] X is a subset of $\mathcal{X}$.

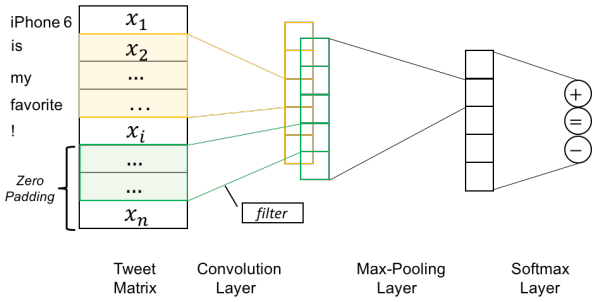

Figure 2: Network structure of the CNN model

A tweet having n words (padded if necessary) was represented as follows:

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \mathbf{x}_3 \oplus ... \oplus \mathbf{x}_n \quad (2)$$

where $\mathbf{x}_i$ is a word vector, and $\oplus$ is the concatenation operator. In general, $\mathbf{x}_{i:i+j}$ meant the concatenation of words $\mathbf{x}_i, \mathbf{x}_{i+1}, ..., \mathbf{x}_{i+j}$.

The layers of the CNN are formed by a convolution operation followed by a pooling operation. We performed a convolution operation to transform a window of h words (i.e., $\mathbf{x}_{i:i+h-1}$) to generate a feature c_i. The procedure was formulated as follows:

$$c_i = \sigma(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b) \quad (3)$$

where $\mathbf{w}$ denotes a filter map, h is the window size of a filter, σ is a non-linear activation function and b is a bias term.

By applying filter $\mathbf{w}$ to each possible window of words in a tweet, we obtained a feature map:

$$\mathbf{c} = [c_1, c_2, ..., c_{n-h+1}] \quad (4)$$

Afterwards, we performed a subsampling operation, for which we used the following max-pooling subsampling method based on the idea of capturing the most important feature from each feature map.

$$c_{max} = max\{\mathbf{c}\} \quad (5)$$

From Eqs. (3)–(5), each filter generated one c_{max} from a tweet matrix.

The number of filter maps in our CNN models was 100, and the possible window sizes were $\{3, 4, and\ 5\}$; thus, our model had 300 different filters in total. The corresponding 300 c_{max} formed the penultimate layer, and was then passed to a fully connected softmax layer to predict the global polarity of a tweet.

4 Experiments

In this section, we compare our deep learning methods with the baseline methods. We first describe our experimental setup, followed by a discussion of the results.

4.1 Experimental Setup

4.1.1 Datasets

The MDSU corpus was originally built for deeper sentiment understanding in a multilingual setting; therefore, tweets in it were annotated many fine-grained tags in addition to global polarity. In this paper, we used global polarities as the classification labels. Lu et al. (2017) filtered out apparent non-emotional tweets and prioritized long tweets with rich language phenomenon during data selection; therefore, the tweets in the MDSU corpus are more complex and longer than those in randomly collected or noisy-labeled tweet datasets.

Table 1 presents the global polarity distribution for each language in the MDSU corpus. The polarity distribution of each language does not differ largely, although not perfectly uniform. Moreover, the polarity distribution of the entire corpus is well-balanced, rendering the corpus suitable for a three-way sentiment classification.

The length of a tweet is defined as the number of elements (including words, emoticons, and punctuations) after under-mentioned pre-processing. The maximum length (also the fixed size of the CNN models) of the MDSU corpus is 124: 41 for English, 93 for Japanese, and 124 for Chinese.

Table 1: Polarity distribution for each language in the MDSU corpus

Language	Abbr.	Positive	Neutral	Negative	Total #	Max Length
English	EN	503	526	774	1803	41
Japanese	JA	392	875	534	1801	93
Chinese	ZH	566	614	638	1818	124
Total	ALL	1461	2015	1946	5422	124

4.1.2 Pre-processing

The language used in social media is more casual than in traditional media. There are many informal ways of expression on Twitter, such as emoticons, Unicode emojis, misspelled words, letter-repeating

words, all-caps words, and special tags (e.g., #, @). These may disturb the learning of word embeddings and classification models; therefore, we pre-processed them.

For all the three languages, we detected Unicode emojis and replaced them with an "EMOJI_CODE" (e.g., we replaced "❤" with "EMOJI_2764"); detected emoticons from easy :-) to complex ((((o(*°▽°*) o)))) using regular expressions and replaced them with "EMOTICON"; and labeled URLs as "URL".

We also performed language-dependent pre-processing. For English, we lowercased English characters and tokenized the tweets with TweetTokenizer[6]; for Japanese, we normalized Japanese characters and tokenized the tweets with Mecab[7]; for Chinese, we transferred traditional Chinese characters to simplified Chinese characters and tokenized the tweets with NLPIR[8].

4.1.3 Word Embeddings and Space Transformation

Large collections of raw tweets are accumulated using Twitter RESTful API by the same query keywords with the MDSU corpus during a one-year period. We excluded undesirable tweets (e.g., tweets starting with "RT") using the same veto patterns as Lu et al. (2017), and checked the preceding 10 tweets of each tweet to delete the repeating tweets. After filtering out these tweets, the remaining tweets were pre-processed as previously described. The number of remaining tweets was 232,214 (EN), 264,179 (JA), and 148,052 (ZH). The vocabulary size for each collection of tweets was 63,343 (EN), 49,575 (JA), and 52,292 (ZH).

Our vector representation for words was learned using fastText[9]. Because the scale of our corpus for word embedding training was relatively small, we set the minimal number of word occurrences as 2. We used the skip-gram model because it generates higher quality representations for infrequent words (Mikolov et al., 2013). The word embeddings for each language were trained separately on its corresponding corpus. Words that were not present in the pre-trained word list were initialized randomly in the deep learning models. The dimension of our word embeddings was 100.

As to the size of bilingual word pairs, it is usually thousands. Figure 3 illustrates how we select bilingual word pairs. First, we translated words in English vocabulary into Japanese/Chinese (using Google Translate), then we selected top K high-frequent English words whose correspondent translations appear in Japanese/Chinese vocabularies. To avoid meaningless words, we filter out those English words whose numbers of characters are less than 2 and that contain any punctuations. In this paper, we set K as $\{1000, 2000, 3000, 4000, 5000\}$ to see how K affects the final performance of MSA.

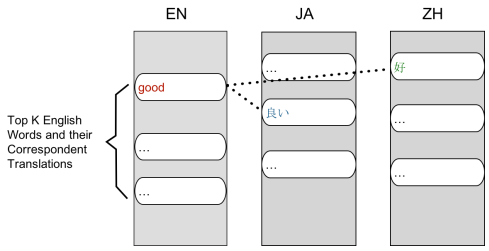

Figure 3: Selection of bilingual word pairs

As a validation, we calculated the change of cosine distance of each word pair in a test set before and after mapping. For each language, a set of another 500 word pairs was held out as the test set (none of them appears in the training set). For simplicity, the two bilingual test sets are denoted as **EN-JP** and **EN-ZH**. Table 2 shows the cosine distance decrease situation in the test sets for the two kinds of transformation methods.

First, we can see that the original angles on average between word pairs were very large for both **EN-JP** and **EN-ZH** (85.2 and 89.3, respectively). As we all know, the closer the angle of a word pair is to 90 degrees, the more irrelevant the word pair is. Therefore, this means that the two vectors of a word pair from separately trained word embeddings has a large difference.

After mapping, the angles became smaller. For *TM*, the angles became 49.5 (averaged over different K, the same below) for **EN-JP**, and 49.6 for **EN-ZH**, respectively; for *OT*, the angles became 63.8 for **EN-JA**, and 60.3 for **EN-ZH**, respectively. This

[6]http://www.nltk.org/api/nltk.tokenize.html
[7]http://taku910.github.io/mecab/
[8]http://ictclas.nlpir.org/
[9]https://github.com/facebookresearch/fastText

shows the transformations both worked. Moreover, the angles of *OT* were larger than that of *TM*. This is reasonable since *OT* imposes additional constraints on *TM*.

As to K, we find that the larger K was, the larger the decrease ratio was for each language and transformation method. This proves that the more training data are used, the better $\mathbf{W}$ is. However, good intermediate results do not necessarily generate good final results. In Section 4.2.3, we will further elaborate which method and K should we use to achieve the best MSA performance.

Table 2: Distance decrease of word pairs in the test sets

Method	Language	Original	k=1000	k=2000	k=3000	k=4000	k=5000
TM	JA	85.2	52.7	50.0	48.9	48.3	47.9
	ZH	89.3	52.7	50.1	49.0	48.4	48.0
OT	JA	85.2	64.8	63.9	63.6	63.4	63.3
	ZH	89.3	61.5	60.5	60.0	59.8	59.6

4.1.4 Model Hyper-parameters

All the methods were evaluated using 10-fold cross validation. For CNN models, we randomly selected 10% of the training splits of cross-validation as the validation datasets to tune parameters for an early stopping. Moreover, trainings were completed using a stochastic gradient descent (SGD) algorithm for shuffled mini-batches with the Adadelta update rule, with a mini-batch size of 50. To prevent overfitting, we employed the dropout technique on the penultimate softmax layers, with a dropout rate of 0.5.

4.2 Result and Discussion

4.2.1 Baselines

Table 3 presents the classification accuracy of baselines.

According to Table 3, the average accuracy of separate SVM classifiers over original datasets was the same as it over translated datasets. This shows that the same method did not necessarily perform worse after being translated by MT for monolingual datasets. In addition, the performance of MT+SVM model (use all translated tweets) was worse than the average accuracy of separate SVM classifiers over original datasets (53.0% vs. 54.5%), showing the limitation of traditional paradigm on MSA (i.e., "MT + machine learning").

For classifiers directly used the cumulation of unigram and bi-gram, both SVM and Banea (2010)* outperformed MT+SVM by 0.8% and 3.4%, respectively. The improvements indicate that the use of cumulation of n-gram is effective, although this may result in the problem of data sparseness (Banea et al., 2010).

Table 3: Accuracies of baselines

Model	Dataset	Feature	Accuracy
Average	–	–	0.545
SVM	EN	unigram+bigram	0.529
SVM	JA	unigram+bigram	0.596
SVM	ZH	unigram+bigram	0.509
Average	–	–	0.545
SVM	EN	unigram+bigram	0.529
SVM	Translated JA	unigram+bigram	0.591
SVM	Translated ZH	unigram+bigram	0.515
MT+SVM (baseline 1)	Translated ALL	unigram+bigram	**0.530**
SVM	ALL	cumulation of unigram+bigram	0.538
Banea (2010)* (baseline 2)	ALL	cumulation of unigram+bigram	**0.564**

4.2.2 Deep Learning Models

Table 4 presents the classification accuracies of the CNN models; the input of word embeddings for the models in this table involved no transformation.

First, our deep learning paradigm performed better than the MT+SVM method (traditional paradigm). Specifically, the parameter-sharing CNN model outperformed MT+SVM model by 4.3% (57.3% vs. 53.0%). This indicates that the deep learning paradigm is more efficient than the traditional paradigm.

Besides, we also conducted the learning separately on each language split. The results revealed that the average accuracy of separate CNN classifiers was a little higher than the accuracy of the mixed case (58.1% vs. 57.3%), implying that the deep learning methods did not improve after using the entire dataset. This is supposed to be caused by the heterogeneity of word embedding spaces, because the raw word embeddings were learned separately.

Furthermore, we observed that the MT+CNN model (trained on the translated datasets and using

only English word embeddings) performed worse than the parameter-sharing CNN model (trained on the original datasets and using multilingual word embeddings). Ideally, if JA/ZH were perfectly translated, the performance should have increased. This suggests that the noises that MT brings in are greater than the heterogeneity of multilingual word embeddings does.

Table 4: Accuracies of deep learning models

Model	Dataset	Accuracy
Average	–	0.581
CNN	EN	0.578
CNN	JA	0.610
CNN	ZH	0.553
MT+CNN	Translated ALL	0.564
Parameter-sharing CNN (none-transformation)	ALL	**0.573**

4.2.3 Deep Learning Models using Transformed Word Embeddings

The coordination of different vector spaces was expected to further improve the deep learning paradigm. Table 5 presents the classification accuracies of the CNN models using differently transformed word embeddings.

According to Table 5, we can see that the performances of *TM* (57.9% on average) were better than none-transformation case, which justified the usefulness of vector space coordination.

Furthermore, *OT* performed ever better than *TM* by 1% on average (58.9%). Artetxe et al. (2016) reported that *OT* could preserve monolingual invariance, which made it perform better in both word analogy and translation induction task. We believe that it is reasonable to attribute the improvement here to this property.

Besides, the performances of both *OT(+LN)* and *OT(+LN&MC)* degraded compared with none-transformation case, especially *OT(+LN)*. This may imply that changing the scale or changing the position and scale together of monolingual word embeddings is unnecessary when carrying out space transformation.

As to K, for both *TM* and *OT* methods, MSA performed best when $K = 1000$. On contrary to the discussion of cosine distance reduction in Section 4.1.3, this suggests that the selection of K isn't 'the more, the better' for down-stream applications (e.g., MSA). What's more, this also justifies the convenience of applying deep learning paradigm to other languages, because we only need to build a small-scale bilingual word pairs.

Overall, the performance of the CNN model fed with orthogonally transformed word embeddings (K=1000) was most effective, which achieved an accuracy of 59.8%.

Table 5: Accuracies of deep learning models using differently transformed word embeddings

Method	k=1000	k=2000	k=3000	k=4000	k=5000	Avg.
TM	0.585	0.578	0.580	0.577	0.574	0.579
OT	**0.598**	0.582	0.588	0.590	0.587	0.589
OT(+LN)	0.532	0.547	0.548	0.542	0.542	0.542
OT(+LN&MC)	0.569	0.559	0.562	0.563	0.568	0.564

5 Conclusion and Future Work

In this paper, we proposed a novel deep learning paradigm for MSA. We map monolingual word embeddings into a shared embedding space, and used parameter-sharing deep learning models to unify the processing of multiple languages. The experiments on a well-balanced tweet sentiment corpus—the MDSU corpus—revealed the effectiveness of our deep learning paradigm. Especially, our CNN model fed with orthogonally transformed word embeddings achieves a rise of 3.4%, comparing with the strong Banea (2010)* baseline.

Our paradigm provides a great cross-lingual adaptability. Tweets in any other language can be represented using transformed word embeddings of that language, and then be channeled into parameter-sharing deep learning models.

The novelty of our study is not in the complexity of the network itself, but more in the coordination of heterogeneous monolingual word embeddings and the parameter-sharing property of the cross-lingual models. In the future, we plan to attempt non-linear transformation methods and more task-oriented deep networks.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, pages 2289–2294.

Alexandra Balahur and Marco Turchi. 2013. Improving sentiment analysis in twitter using multilingual machine translated data. *In Proceedings of Recent Advances in Natural Language Processing*, pages 49–55.

Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2010. Multilingual subjectivity: Are more languages better? *In Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 28–36.

Kerstin Denecke. 2008. Using sentiwordnet for multilingual sentiment analysis. *In Proceedings of the 24th International Conference on Data Engineering Workshop (ICDE 2008)*, pages 507–512.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report (Stanford)*, pages 1–6.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Yujie Lu, Kotaro Sakamoto, Hideyuki Shibuki, and Tatsunori Mori. 2017. Construction of a multilingual annotated corpus for deeper sentiment understanding in social media. *Journal of Natural Language Processing*, 24(2):205–266.

Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. *In Proceedings of the 45th Annual Meeting of the As-sociation of Computational Linguistics (ACL 2007)*, pages 976–983.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv:1309.4168*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. *Proceedings of Empirical Methods on Natural Language Processing (EMNLP 2002)*, pages 79–86.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. Insight-1 at semeval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis. *In Proceedings of Proceedings of SemEval-2016*, pages 330–336.

Sebastian Ruder. 2017. a survey of cross-lingual embedding models. *arXiv:1706.04902*.

Josef Steinberger, Polina Lenkova, Mijail Kabadjov, Ralf Steinberger, and Erik van der Goot. 2011. Multilingual entity-centered sentiment analysis evaluated by parallel corpora. *Proceedings of Recent Advances in Natural Language Processing*, pages 770–775.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1815–1827.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. *In Proceedings of the Joint Conference of the 47th An-nual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243.