On the Reproducibility and Generalisation of the Linear Transformation of Word Embeddings

Xiao Yang[1], Iadh Ounis[1], Richard McCreadie[1], Craig Macdonald[1], and Anjie Fang[2]

University of Glasgow, Glasgow, UK
[1] firstname.lastname@glasgow.ac.uk, [2] a.fang.1@research.gla.ac.uk

Abstract. Linear transformation is a way to learn a linear relationship between two word embeddings, such that words in the two different embedding spaces can be semantically related. In this paper, we examine the reproducibility and generalisation of the linear transformation of word embeddings. Linear transformation is particularly useful when translating word embedding models in different languages, since it can capture the semantic relationships between two models. We first reproduce two linear transformation approaches, a recent one using orthogonal transformation and the original one using simple matrix transformation. Previous findings on a machine translation task are re-examined, validating that linear transformation is indeed an effective way to transform word embedding models in different languages. In particular, we show that the orthogonal transformation can better relate the different embedding models. Following the verification of previous findings, we then study the generalisation of linear transformation in a multi-language Twitter election classification task. We observe that the orthogonal transformation outperforms the matrix transformation. In particular, it significantly outperforms the random classifier by at least 10% under the F1 metric across English and Spanish datasets. In addition, we also provide best practices when using linear transformation for multi-language Twitter election classification.

Keywords: embedding, linear transformation, Twitter classification

1 Introduction

Word embeddings are particularly useful as text representations, since semantically (rather than textually) similar words can be found using similarity metrics (e.g. *cosine* similarity) [1]. Therefore, there is an increasing interest in using multilingual word embeddings to capture semantic similarities among different languages. For example, recent works have learned multilingual word embeddings using monolingual text corpora along with a parallel corpus of aligned words and/or sentences [2–4]. Based on the observation that similar words in different languages have similar geometric arrangements in word embedding spaces, Mikolov et al. [1] showed that multilingual word embeddings can be obtained "offline" using a *linear transformation*. Despite the simplicity of linear transformation, it has been shown to be effective for machine translation, i.e. when aiming to translate words from a source language to another language. Using a large scale training dictionary of more than 10^{10} English and Spanish word pairs, a linear transformation approach achieved 0.53 precision@1 [1].

Furthermore, recent enhancements have been proposed to make linear transformation more effective, namely: by retrieving translation pairs [5]; or learning a linear transformation matrix based on orthogonal transformation (e.g. by leveraging canonical correlation analysis (CCA) [6, 7] or singular value decomposition (SVD) [8–10]). Given the current research interest in the use of word embeddings in various tasks, such as information retrieval [11–13] and text classification [14–16], the reproduction, validation, and generalisation of findings from the literature of linear transformation are important for extending that research for multilingual scenarios. As such, in this paper, we examine the reproducibility and generalisation of linear transformation of word embeddings in different languages.

We begin by reproducing two previous linear transformation approaches:

1. Matrix transformation (denoted `MT`) proposed by Mikolov et al. [1]
2. Orthogonal transformation that uses SVD (denoted `OT`) [9].

We choose these two approaches because, to the best of our knowledge, `MT` is the first attempt to address linear transformation of word embedding, while `OT` is a recent approach that claims to provide better performance over previous approaches. Over a simple machine translation task using our own word-aligned translation corpus of English and Spanish words, we validate the consistency and performance of linear transformation. We also evaluate the generalisation of linear transformation by applying it to a multi-language Twitter election classification task that classifies each tweet as "election-related" or "other". This task aims to adapt or transfer an existing classifier trained on a Twitter election dataset in English to that of Spanish and vice versa. This is particularly useful in monitoring emerging topics during the lead-up to an election, where well-designed training/test collections are not available. Our results on 3 Twitter election datasets (in two different languages) show that linear transformation is generalisable to the multi-language Twitter election classification task.

The remainder of this paper is organised as follows: We first describe the linear transformation approaches in Section 2. We report our experimental setup in Section 3, describing the datasets we used, classifier and the evaluation process. In Section 4, we present the results of a simple machine translation task, validating the reproducibility of linear transformation. In Section 5, we study the generalisation of linear transformation and present results for multi-language Twitter election classification. Finally, Section 6 summarises our conclusions.

2 Linear Transformation

Linear transformation approaches, for example the matrix transformation (`MT`) approach [1] and the orthogonal transformation approach that uses SVD (`OT`) [9], allow the transfer of pre-trained monolingual embedding models "offline" using aligned words in two languages. In particular, experiment using the orthogonal transformation approach (OT) demonstrate that a linear mapping between embedding spaces should be orthogonal to achieve enhanced performance [8, 9]. In the rest of this section, we detail the implementation of the 2 approaches we used.

MT **Approach [1]:** In this approach, a list of word pairs $\{x_i, y_i\}_{i=1}^{n}$ is generated by using *Google Translate*, where y_i is the translation of x_i. Word x_i in source language is extracted from a background text corpora (e.g. comprised of Google News articles). As such, words x_i and y_i have the same meaning but in two different languages. Then, a linear matrix W is trained by using gradient descent to minimise the squared reconstruction error, as shown in:

$$\underset{W}{min} \sum_{i=1}^{n} |y_i - W x_i|^2 \tag{1}$$

After the training process, one word vector can be projected to a vector in another space by applying $y'_i = W x_i$. To find a similar word in another space, one can simply use cosine similarity to find the translation of x_i, whose vector is the closest to y'_i.

OT **Approach [9]:** Smith et al. [9] provided an enhanced version of MT based on orthogonal transformation. When matrix W maps one embedding space A to the embedding space B, W^T should be able to map the embedding space B back to embedding space A, i.e. we have $y \sim Wx$ and $x \sim W^T y$. This means that the transformation matrix W is supposed to be an orthogonal matrix O with $O^T O = I$, where I is the identity matrix. Therefore, using this orthogonal matrix O, one can obtain a word similarity matrix $S = YOX^T$, where $S_{i,j} = |y_i||Ox_j|cos(\theta_{i,j}) = cos(\theta_{i,j})$ if X and Y are normalised. Note that matrix S contains the similarity of any word pairs from the embedding spaces of the two languages. Similarly, an orthogonal matrix is trained by maximizing the similarity of the ground truth word pairs $\{x_i, y_i\}_{i=1}^{n}$. This process is shown in the following equation:

$$\underset{O}{max} \sum_{i=1}^{n} y_i^T O x_i, \text{where } O^T O = I \tag{2}$$

To implement the training process, vectors of words in $\{x_i, y_i\}_{i=1}^{n}$ (denoted as $\{X_D, Y_D\}$) are first retrieved from their embedding spaces, respectively. Next, a singular value decomposition (SVD) is applied following $M = Y_D^T X_D = U \varSigma V^T$, where U and V are made up of the orthonormal vectors and $\varSigma$ contains singular values. The optimised similarity matrix can be obtained as follows:

$$S = YUV^T X^T, \text{where } S_{i,j} = y_i^T U V^T x_j = (U^T y_i) \cdot (V^T x_j) \tag{3}$$

Therefore, both embedding spaces can be mapped into a single space by applying V^T to X and U^T to Y. In this paper, we use the MT implementation of Dinu et al. [5], which solves Eq. (1) using least squares[1]. For OT, we directly use the codes from Smith et al. [9], which is publicly available[2].

[1] clic.cimec.unitn.it/georgiana.dinu/down/

[2] github.com/Babylonpartners/fastText_multilingual

3 Experimental Setup

In this section, we briefly describe the word embedding models, as well as provide details about the evaluation datasets and metrics used in our experiments.

3.1 Word Embeddings

For the purpose of reproducibility, we use pre-trained and publicly available word embedding models instead of training our own models. The publicly available word embedding models were trained using *fastText*[3] from Wikipedia corpora since *fastText* has proved to be both effective and efficient [17]. We only choose the English and Spanish embedding models from 294 available languages[4], as these are the languages of our Twitter election datasets. In particular, these embedding models have 300 dimensions and are obtained using the skip-gram model with default *fastText* settings.

3.2 Translation Corpus

In order to learn and test the transformation matrix of `MT` and `OT`, a translation corpus is required to provide word-level alignment of the two languages. We also use the translation corpus to reproduce the translation task in [1, 9]. Each word alignment is a pair of a Spanish word and its translation in English. We extract the most common 50k words from a Spanish Wikipedia snapshot dated 02/10/2015, excluding stopwords (e.g. "un", "es", "yo" and etc.). Afterwards, their corresponding English translations are obtained using the *Google Translate* service. Due to the nature of languages, the English translations may contain multiple words (e.g. "lanzado" is translated as "thrown out"). Indeed, 3,817 such translations are not considered as word-level alignments in this paper. In addition, *Google Translate* fails to translate 14,504 words extracted from the Spanish Wikipedia snapshot (e.g. "lobería", "porrón" and "ciénega"). These cases are removed from the translated corpus. We choose Wikipedia as the source to obtain translation pairs since the publicly available word embedding models from *fastText* are trained on Wikipedia corpora. As such, we can minimise occurrences of the out-of-vocabulary (OOV) problem when training and testing the linear transformation matrix. The Spanish Wikipedia snapshot (dated 02/10/2015) we use contains $1.15M+$ documents and about $436K$ unique words excluding stopwords. The final translation corpus consists of 29,907 Spanish-English word pairs.

3.3 Twitter Election Datasets

To evaluate the generalisation of linear transformation on a multi-language Twitter election classification task, we use 3 Twitter election datasets[5] in this paper.

[3] github.com/facebookresearch/fastText

[4] github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[5] dx.doi.org/10.5525/gla.researchdata.564

Venezuela Election. We target the 2015 Venezuela parliamentary election, which was held on the 6^{th} December 2015 to elect the 164 deputies and 3 indigenous representatives of the National Assembly.

Philippines Election. We target the 2016 Philippines general election, which was held on the 9^{th} May 2016 for the executive and legislative branches of all levels of government. The Philippine presidential and vice-presidential elections of 2016 were held as part of the general election, and are covered in this dataset.

Ghana Election. We target the 2016 Ghana general election, which was held on the 7^{th} December 2016 to elect a president and members of parliament.

Before collecting Twitter posts about each election, we had political science experts selected a number of keywords (e.g. PHVote and GHElection) and Twitter user accounts (e.g. election candidates and news media) by browsing the Twitter posts related to a given election. We then use the Twitter Streaming API to collect Twitter posts that contain either one of the specified keywords or that are posted by one of the selected Twitter user accounts. In addition, we only collect Twitter posts that were published during the period of one month before and after the election date since this period potentially covers more relevant pre- and post election topics.

Since millions of tweets were collected for each target election, we adopt the classical TREC-style pooling methodology [18] that will be described later. This allows human assessors to identify election-related tweets without having to judge all of the tweets. Moreover, we allow our political science experts to suggest queries (keywords related to the election e.g. election, vote buying and supporters clash), and use the Terrier IR platform [19] to rank the retrieved tweets of each query per day. When ranking tweets, we use the DFReeKLIM [20] weighting model, which is designed for the effective retrieval of short documents like tweets. Finally, only the top-ranked 7 tweets for each query per day are added to the *pool* of tweets to be assessed because this gives a tweet collection of approximately 4k – 5k tweets, which allows our human annotators to finish the annotation job in a short time. Each sampled tweet is labelled as: "election-related" or "other" by our 5 political science experts. The final label of a tweet is then determined by a majority vote. Overall, for Venezuela, Philippines and Ghana datasets, we found moderate agreements of 52%, 68% and 71% respectively between all assessors using Cohen's *kappa*. The general statistics of our datasets such as the dominant language and the number of tweets in each category are shown in Table 1. The datasets cover two languages: English and Spanish, which are used to evaluate the performance of linear transformation in the multi-language Twitter election classification task.

Using the generated election datasets, we consider two settings in this paper: Train a classifier on an English election dataset A and test the classifier on a Spanish election dataset B (denoted A $\Rightarrow$ B), and vice versa (denoted B $\Rightarrow$ A). We also split our election datasets into different subsets for each setting. For example, for A $\Rightarrow$ B, 60% of instances are randomly sampled from dataset A as D_s to train classifiers and the remaining 40% in dataset A as validation set D_s^v. 90% of instances from dataset B are sampled as the out-of-sample D_t^o that is used as the test set to evaluate the performance in another election; the

Table 1. Statistics of the Twitter election classification datasets.

Election	Language	Election-related	Other	Total
Venezuela	Spanish	2,273	3,474 (60%)	5,747
Philippines	English	1,755	2,408 (58%)	4,163
Ghana	English	1,254	1,999 (61%)	3,253

remaining 10% (D_t^v) in the dataset B is used to track the performance of linear transformation during the training of the classifiers.

3.4 Classifier

In order to study the generalisation of linear transformation on the multi-language Twitter election classification task and evaluate its performance, we need to learn a text classifier on the training dataset in one language and apply it to a test dataset in another language. A variety of learning algorithms are available for such a task, such as random forest and support vector machines (SVM). However, one of the most recent and effective algorithms is based upon *Convolutional Neural Networks* (CNN) [15, 16]. CNN classifiers have shown their effectiveness for Twitter classification tasks, such as sentiment analysis [21, 22]. In addition, CNN can work with word embeddings by simply stacking the word vectors. Through the convolution operations, local indicators that are important for the classification task can be learned from the labelled dataset by sliding filters over the vector features. Therefore, in this paper, we use CNN classifiers with word embeddings to evaluate the classification performance of linear transformation. In particular, we train CNN classifiers on a training dataset D_s and then test it on a test dataset D_t^o that is in another language. The words in the test datasets are transformed into the embedding space we used to train the CNN classifiers. When transforming a word from a source language to the target language, the transformation matrix W is applied to its word vector x_i. As such, the transformed vector $y_i' = Wx_i$ is used as part of the vector representations of a tweet. Such representations can then be used by the CNN classifier to classify "election-related" tweets in the unseen test dataset. Furthermore, a regularisation technique, namely dropout, is also applied to the CNN to only keep a neuron active with some probability p during training [15]. To evaluate the effectiveness of linear transformation, `MT` and `OT` are compared with a random baseline that makes predictions randomly according to the distribution of election-related tweets in the training datasets.

3.5 Training, Hyper-parameters & Metrics

To evaluate the performance in the translation task, for consistency, we use the same metrics that are used by Mikolov et al. [1] and Smith et al. [9], namely: precision@1 (P@1), precision@5 (P@5) and precision@10 (P@10). These three metrics evaluate how many words in the test translation corpus have the correct translations in the retrieved top k translations ranked by the cosine similarity.

Table 2. Translation results using *pseudo-dictionary* with various dictionary sizes. Best scores are highlighted in bold.

Training Set Size	Algorithm	English to Spanish			Spanish to English		
		P@1	P@5	P@10	P@1	P@5	P@10
∼ 5k	`MT`	0.014	0.040	0.062	0.021	0.051	0.072
	`OT`	0.264	0.471	0.537	0.407	0.567	0.611
∼ 250k	`MT`	0.010	0.041	0.064	0.033	0.062	0.083
	`OT`	0.345	0.562	0.625	0.505	0.651	0.685
∼ 608k	`MT`	0.010	0.039	0.062	0.032	0.063	0.084
	`OT`	**0.348**	**0.566**	**0.628**	**0.509**	**0.654**	**0.689**

For the multi-language Twitter election classification task, we report the precision, recall and F1 score. We set up classifiers with filter size $m = 1$ and dropout rate $p = 0.5$. We pad short tweets to the length of the longest tweet using a special token, which are initialised as zero vectors.

4 Reproducibility – Linear Transformation Performance

This section reports our attempts to reproduce the results presented in the recent linear transformation paper [9] that uses SVD based orthogonal transformations. In this paper, we reproduce the results of the simple machine translation task, which attempts to retrieve the correct translation of a given word in a source language. The word in a source language is transformed into the target language using the linear transformation mentioned in Section 2. Then the transformed vector is used to retrieve the closest word in the target language by cosine similarity. In previous work, linear transformation has been evaluated in the translation task using an English-Italian translation corpus [5, 8, 9]. Thus, we use a different translation corpus of Spanish-English to validate whether the previous findings can be reproduced. We sample 1,000 Spanish-English translation pairs from our translation corpus as the translation test set, while using the rest as the translation training set. To reproduce previous findings in [9], we also include a *pseudo-dictionary* as another translation training set, which consists of identical character strings shared by both Spanish and English word embedding models. 608,772 such identical words appear in both embedding models, including loanwords from the two languages such as “TV”, “IBM” and “fanatica”. However, such identical word pairs in two languages do not necessarily have the same meaning, e.g. “once” is written identically in English and Spanish but has different meanings in each language. Moreover, in addition to experiments originally performed in [9], we also vary the size of the translation training set to evaluate the performance in different sizes.

In Table 2, we first present the translation performance using the pseudo-dictionary. We train the orthogonal transformation matrix W and transform the source embedding space to the target embedding space. Afterwards, we predict translations of words in the source embedding by a nearest neighbour retrieval as detailed by Mikolov et al. [1].

Table 3. Translation results using our *translation corpus* with various dictionary sizes. Best scores are highlighted in bold.

Training Set Size	Algorithm	English to Spanish			Spanish to English		
		P@1	P@5	P@10	P@1	P@5	P@10
$\sim$ 5k	MT	0.446	0.652	0.712	0.590	0.752	**0.793**
	OT	0.464	0.669	0.726	0.604	0.756	0.789
$\sim$ 15k	MT	0.430	0.628	0.698	0.577	0.746	0.783
	OT	0.466	**0.678**	**0.732**	0.615	**0.760**	0.789
$\sim$ 25k	MT	0.442	0.624	0.681	0.568	0.733	0.777
	OT	**0.469**	0.675	0.729	**0.616**	0.758	0.785

In particular, we vary the size of the pseudo-dictionary to validate the effect of the dictionary size. We randomly sample 5k, 250k and 608k pairs without replacement from the entire pseudo-dictionary to train the transformation matrix W. At the end, the learned transformation matrix is applied and evaluated on the test set of our translation corpus. From Table 2, we see clearly that OT outperforms MT in both translations from English to Spanish and from Spanish to English. On our test dictionary set, OT achieved best P@1 of 0.348 and 0.509 for English to Spanish and Spanish to English respectively. However, MT only achieved 0.01 and 0.032 respectively. This validates the previous finding of Smith et al. [9], and shows the advantage of orthogonal transformations over the original linear transformation proposed by Mikolov et al. [1]. In addition to the experiments in [9], by increasing the size of training dictionary, our results show that both OT and MT can slightly improve their performance, however the improvement is minimal when the size is greater than 250k.

In Table 3, we show the translation performance using the training dictionary from our translation corpus. Similar to the experiment on the pseudo-dictionary, we vary the size of the training dictionary. However, we split the traning dictionary based on the word frequecy in the Spanish Wikipedia snapshot we used. Compared with the results of using a pseudo-dictionary, it is unsurprising that the performance is much better for both OT and MT since the quality of aligned dictionary is better than the pseudo-dictionary. This also validates the results in [9] that an accurate translation dictionary is important for learning an effective translation matrix. In particular, MT shows comparable performance with OT on P@5 and P@10. However, in our additional exepriments, we show that increasing the size of the training dictionary does not lead to an improvement on P@5 and P@10 after 15k for both OT and MT.

To provide insights on the difference between MT and OT, we show the two dimensional PCA projections of sampled words in Figure 1. Using samples from the pseudo-dictionary, we show that source words (in English) transformed by OT are generally closer to the corresponding target words (in Spanish). The samples from the aligned dictionary show that OT has better performance since the transformed words are closer to the target language than MT. Overall, we reproduced results and findings of the previous work [9], which shows that OT is more effective in the translation tasks, even when trained on a pseudo-dictionary.

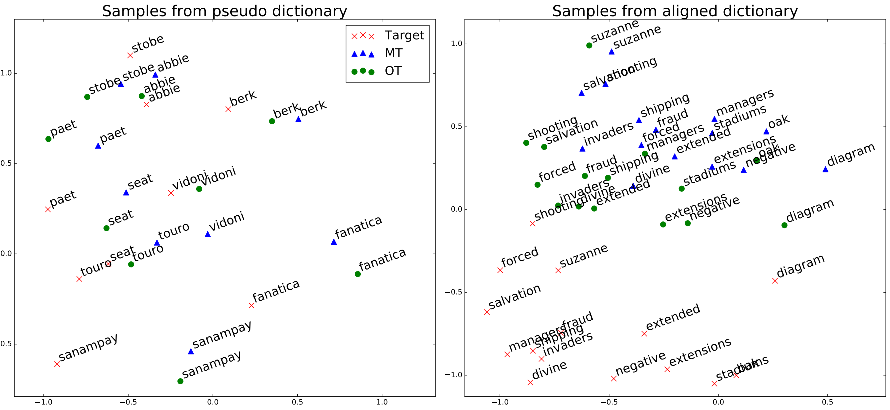

Fig. 1. Two dimensional PCA projections of words sampled from pseudo and aligned dictionaries. The target language is Spanish (translated in this figure). English words are transformed to the Spanish word embedding using `MT` and `OT`.

Furthermore, we observe that by increasing the size of the training set, improvements in translation performance rapidly diminish.

5 Generalisation – Multi-language Twitter Classification

By applying linear transformation to a multi-language classification task, we report the generalisation of linear transformation. The obtained results are shown in Table 4 and 5 where the first column shows the algorithms, and other columns show the classification performance by scenario, for example "Philippines ⇒ Venezuela" shows the results of training a classifier on the Philippines dataset (D_s) and testing on the Venezuela dataset (D_t^o). When testing the classifier on a test dataset, we use linear transformation to translate the corresponding word embedding vectors into the embedding space we used to train the classifiers.

As shown in Table 4, the classifier trained using `OT` indeed outperforms `MT` in the "Venezuela ⇒ Philippines" scenario. However, by training classifiers on the Philippines dataset and testing on the Venezuela dataset, `MT` shows a slightly better performance under all of the metrics tested. In particular, only `OT` outperforms the `Random` classifier in both of the two classification tasks, which shows that `OT` can better capture linear transformation between the two languages. We note that, by transforming Spanish embeddings into English embeddings, better performance can be achieved. Such an observation is similar to that of the translation task we examined in Section 4. In Table 5, we evaluate the performance between the Ghana dataset and the Venezuela dataset. In both of the two classification tasks, the F1 score of `OT` outperforms `Random` and `MT`. Compared with the results in Table 4, the overall performance drops for `MT` and `OT` when tested on the Venezuela dataset. Many factors may lead to such a performance drop. For example, depending on the election period and candidates, word distributions can vary in different elections. In addition, by simply translating a word

Table 4. Classification results using transformed embedding models. † indicates significant improvement over random classifier.

Algorithm	Philippines ⇒ Venezuela			Venezuela ⇒ Philippines		
	Precision	Recall	F1	Precision	Recall	F1
Random	0.399	0.418	0.409	0.417	**0.379**	0.398
`MT`	**0.612**	**0.785**	**0.688**†	**0.956**	0.195	0.324
`OT`	0.608	0.783	0.684†	0.949	0.322	**0.481**†

Table 5. Classification results using transformed embedding models. † indicates significant improvement over random classifier.

Algorithm	Ghana ⇒ Venezuela			Venezuela ⇒ Ghana		
	Precision	Recall	F1	Precision	Recall	F1
Random	0.399	**0.418**	0.409	0.397	0.390	0.394
`MT`	0.788	0.270	0.402	**0.916**	0.116	0.206
`OT`	**0.793**	0.387	**0.520**†	0.890	**0.450**	**0.598**†

into another language, it neglects the word order in different languages. Another factor is that the size of the Philippines dataset is larger than the Ghana dataset, therefore it may have better overlap with the Venezuela dataset on election topics. These factors can all affect the classification performance, which shows the complexity of the multi-language classification when compared with the simpler translation task examined previously in Section 4. In addition, we observe that `MT` performs poorly performance when applied in the "Venezuela ⇒ Philippines" and "Venezuela ⇒ Ghana" scenarios. Overall, only `OT` achieved significant improvements over the random classifier on all the aforementioned classification tasks, which yields p-value < 0.05 using McNemar's test.

When training classifiers, we track the performance of the linear transformation models when training each classifier. The performances over the training steps are shown in Figure 2 where we track the F1 scores of classifiers on the validation set D_s^v of the training dataset (shown as "Validation"), on the subset D_t^v of the test dataset (in another language) using two approaches `OT` and `MT`. As shown in Figure 2, the performances of both `OT` and `MT` improve at the beginning of the training process until the performance of classifiers converges on the classification training dataset. However, the performance of `MT` tends to drop when continuing to train the classifiers. In contrast, `OT` is more stable and able to retain the attained performance along the entire training process. In particular, in the task "Venezuela ⇒ Ghana", `OT` tends to improve the performance continuously while the performance of `MT` decreases dramatically. The diverging behaviours of `MT` and `OT` in Figure 2 shows that `OT` can map the relationship of two embedding spaces better than that of using `MT`. Therefore, when training classifiers, `OT` is less sensitive to the new batches of training instances. Additionally, our results show that, as a best practice, stopping the training process of classifiers earlier when the classification performance converges can potentially help `MT` avoid a decline in performance by 10% to 20% F1.

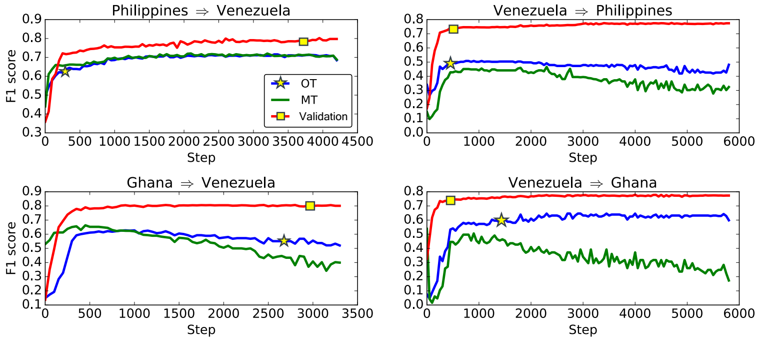

Fig. 2. Performance of linear transformation along the training process of classifiers. Square and star signs are used to distinguish the lines of Validation, `OT` and `MT`.

6 Conclusions

In this paper, we have reproduced, validated, and generalised findings of linear transformation of word embeddings from the literature. We evaluated linear transformation approaches on two different tasks (e.g. machine translation and multi-language Twitter election classification), making further observations from our experiments. In conclusion, we have confirmed that the orthogonal transformation using SVD [9] indeed outperforms the original approach proposed by Mikolov et al. [1] in Section 4. In particular, the orthogonal transformation can still learn a reasonable transformation matrix only using a pseudo-dictionary that contains words shared by the embedding models. Moreover, we show that by increasing the size of the training dictionary set, further gains in translation performance rapidly diminish. Furthermore, in Section 5, we apply linear transformation approaches to a multi-language Twitter election classification task, which is a more complex task than the translation task commonly examined in the literature. We observe that again the orthogonal transformation is more effective in all the classification scenarios than the original approach. Moreover, its performance is significantly better than a random classifier with at least 10% improvement in F1 score, thus we show the effectiveness of linear transformation without any prior knowledge from the test dataset (in another language). We also showed that a best practice is to halt the training process the classifier when convergence is reached, as this can potentially avoid a performance drop off. Finally, given that the performance of linear transformation varies on different datasets, we conclude that future work should investigate what are the factors that affect the translation and classification performance and how to leverage on these factors to improve linear transformation.

Acknowledgements

This paper was supported by a grant from the Economic and Social Research Council, (ES/L016435/1). The authors would like to thank the assessors for their efforts in reviewing tweets.

References

1. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013)
2. Chandar, S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V.C., Saha, A.: An autoencoder approach to learning bilingual word representations. In: Proc. of NIPS. (2014)
3. Eger, S., Hoenen, A.: Language classification from bilingual word embedding graphs. arXiv preprint arXiv:1607.05014 (2016)
4. Zhou, H., Chen, L., Shi, F., Huang, D.: Learning bilingual sentiment word embeddings for cross-language sentiment classification. In: Proc. of ACL. (2015)
5. Dinu, G., Lazaridou, A., Baroni, M.: Improving zero-shot learning by mitigating the hubness problem. arXiv preprint arXiv:1412.6568 (2014)
6. Ammar, W., Mulcaire, G., Tsvetkov, Y., Lample, G., Dyer, C., Smith, N.A.: Massively multilingual word embeddings. arXiv preprint arXiv:1602.01925 (2016)
7. Faruqui, M., Dyer, C.: Improving vector space word representations using multilingual correlation. In: Proc. of EACL. (2014)
8. Artetxe, M., Labaka, G., Agirre, E.: Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In: Proc. of EMNLP. (2016)
9. Smith, S.L., Turban, D.H.P., Hamblin, S., Hammerla, N.Y.: Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In: Proc. of ICLR. (2017)
10. Xing, C., Wang, D., Liu, C., Lin, Y.: Normalized word embedding and orthogonal transform for bilingual word translation. In: Proc. of HLT-NAACL. (2015)
11. Mitra, B., Nalisnick, E., Craswell, N., Caruana, R.: A dual embedding space model for document ranking. arXiv preprint arXiv:1602.01137 (2016)
12. Moran, S., McCreadie, R., Macdonald, C., Ounis, I.: Enhancing first story detection using word embeddings. In: Proc. of ACM SIGIR. (2016)
13. Fang, A., Macdonald, C., Ounis, I., Habel, P., Yang, X.: Exploring time-sensitive variational bayesian inference lda for social media data. In: Proc. of ECIR. (2017)
14. Yang, X., Macdonald, C., Ounis, I.: Using word embeddings in Twitter election classification. In: Proc. of Neu-IR workshop at SIGIR. (2016)
15. Kim, Y.: Convolutional neural networks for sentence classification. In: Proc. of EMNLP. (2014)
16. Severyn, A., Nicosia, M., Barlacchi, G., Moschitti, A.: Distributional neural networks for automatic resolution of crossword puzzles. In: Proc. of IJCNLP. (2015)
17. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
18. Voorhees, E.M., Harman, D.K.: TREC: Experiment and Evaluation in IR. MIT Press (2005)
19. Macdonald, C., McCreadie, R., Santos, R.L., Ounis, I.: From puppy to maturity: Experiences in developing Terrier. In: Proc. of OSIR workshop at SIGIR. (2012)
20. Amati, G., Amodeo, G., Bianchi, M., Marcone, G., Bordoni, F.U., Gaibisso, C., Gambosi, G., Celi, A., Di Nicola, C., Flammini, M.: FUB, IASI-CNR, UNIVAQ at TREC 2011 microblog track. In: Proc. of TREC. (2011)
21. Severyn, A., Moschitti, A.: UNITN: Training deep convolutional neural network for Twitter sentiment classification. In: Proc. of SemEval. (2015)
22. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for Twitter sentiment classification. In: Proc. of ACL. (2014)