Transfer Learning for Multi-language Twitter Election Classification

Xiao Yang Richard McCreadie Craig Macdonald Iadh Ounis
firstname.lastname@glasgow.ac.uk
University of Glasgow, UK

Abstract—**Both politicians and citizens are increasingly embracing social media as a means to disseminate information and comment on various topics, particularly during significant political events, such as elections. Such commentary during elections is also of interest to social scientists and pollsters. To facilitate the study of social media during elections, there is a need to automatically identify posts that are topically related to those elections. However, current studies have focused on elections within English-speaking regions, and hence the resultant election content classifiers are only applicable for elections in countries where the predominant language is English. On the other hand, as social media is becoming more prevalent worldwide, there is an increasing need for election classifiers that can be generalised across different languages, without building a training dataset for each election. In this paper, based upon transfer learning, we study the development of effective and reusable election classifiers for use on social media across multiple languages. We combine transfer learning with different classifiers such as Support Vector Machines (SVM) and state-of-the-art Convolutional Neural Networks (CNN), which make use of word embedding representations for each social media post. We generalise the learned classifier models for cross-language classification by using a linear translation approach to map the word embedding vectors from one language into another. Experiments conducted over two election datasets in different languages show that without using any training data from the target language, linear translations outperform a classical transfer learning approach, namely Transfer Component Analysis (TCA), by 80% in recall and 25% in F1 measure.**

I. Introduction

Social media platforms, such as Twitter, have been widely used in reporting news and events, including a diverse range of political insights and commentary [1], [2]. Therefore, social scientists are interested in mining social media data to study public opinions, election topics, electoral malpractice and violence [3], [4]. For example, social scientists can monitor electoral violence from related Twitter posts published by news agents or citizens in real-time. However, due to the large numbers of tweets, it is important to deploy automatic supervised approaches to identify tweets about each election.

In general, traditional machine learning approaches assume, in classification tasks, that the domains (e.g. movie reviews) of the training and test sets are identical [5]. However, in practice, elections around the world are often reported and discussed in different languages since the official or *de facto* languages differ between countries. For example, a classifier trained on social media posts concerning the Venezuela election (in Spanish) may not be effective when applied to posts from the Philippines election (mostly in English), because the main languages used in these two countries are different. Although individual classifiers can be trained for each independent election, it is time-consuming to create numerous well-designed training/test collections, due to the necessary and expensive human labelling efforts. Moreover, political scientists are interested in monitoring emerging topics during the lead-up to an election. However, representative data collection may not be available from the start of that election. Therefore, it would be beneficial if we could *transfer* an existing election classifier trained for a language A, and adapt it for use on a new election in a different language B.

Transfer learning techniques are a means to bridge the feature gap when changing the domain of classification. For instance, transfer learning approaches have been used previously for generalising classifier training examples into other languages [6]. Meanwhile, recent advances in neural network learning now allow for word embeddings to be learned via shallow neural networks, enabling the more effective modelling of relations between words [7]. Using neural networks, recent works have proposed approaches to learn cross-lingual word embeddings [8]–[10] from multilingual corpora for tasks such as cross-language sentiment classification. For instance, Mikolov et al. [11] proposed a *linear translation* approach based on the observations that similar words in different languages have similar geometric arrangements in word embedding spaces. Therefore, for a classification task, word embeddings can potentially be used to bridge the feature gap caused by languages.

However, the best practices when combining transfer learning and word embeddings are not yet well understood. There are a series of experimental factors that can affect the generalization of the resultant model, such as the pre-processing steps applied to the posts, how to train the translation matrix, and the transfer learning approach used. Hence, as a step towards better understanding of how to build cross-lingual classification models, in this paper we study approaches that combine transfer learning with word embeddings for the task of identifying election-related content on social media.

The contributions of this paper are three-fold. First, we show that when working with Twitter data, applying text preprocessing (e.g. replacing the Twitter handles and hashtags with the generalization words “mention” and “hashtag”) leads to more generalizable models. Second, when training the word

ASONAM '17, July 31-August 03, 2017, Sydney, Australia
© 2017 ACM ISBN 978-1-4503-4993-2/17/07...$15.00
http://dx.doi.org/10.1145/3110025.3110059

embeddings translation matrix, we show that there is little gain to be found by integrating a larger size of translation corpus with the pre-trained word embedding models. Finally, our results show that for performing transfer learning, using linear translations outperforms the popular Transfer Component Analysis approach when applied to convolutional neural networks (CNN) and support vector machines (SVM).

In the remainder of this paper, we briefly discuss the related work in Section II. The linear translation approach we used to transform word embedding vectors between different languages is introduced in Section III. We define our experimental setup in Section IV and describe our experimental results in Section V. Concluding remarks are provided in Section VI.

II. Related Work

In this section, we briefly introduce recent and related work in the areas of text classification, word embeddings, cross-language classification, as well as discuss how they relate to the study presented in this paper.

Text Classification: Supervised text classification is a common task in Information Retrieval (IR). It involves training a model for distinguishing documents (e.g. tweets) into different categories based on the features of those documents. In particular, a supervised learning algorithm takes an input as series of <feature-vector,class> pairs (known as instances). One goal of the learning algorithm is to find a combination of the features that results in predicting the correct class with as little error as possible. A variety of learning algorithms have been proposed in the literature, such as linear regression, J48 trees and support vector machines (SVM). However, one of the most recent effective algorithms is based upon *Convolutional Neural Networks* (CNN) [12], [13]. Leveraging the convolution operation, important local indicators can be learned from the labelled dataset by sliding filters over the vector features of each instance. To avoid overfitting, a regularization technique, namely dropout, is often applied to CNN to only keep a neuron active with some probability p during training [12]. CNN has shown its effectiveness in different Twitter classification tasks [14]–[16]. As such, in this paper, we use CNN to learn tweet classification models in addition to SVM.

Word Embeddings: In most text classification tasks, the presence of terms within the document are used as features. However, *word embeddings* have emerged as an effective alternative representation to using classical TF-IDF vector representations, for example in Twitter sentiment classification tasks [14], [15], [17]. Word embedding is a technique for learning word vector representations of text from a shallow neural network using a background corpus [7], [18], [19]. Each word vector represents the geometric location of a corresponding word in the embedding space and similar words are close to each other in that space. Word embeddings have recently become popular as a text representation, since the vectors produced can be compared to find semantically (rather than textually) similar words using similarity metrics (e.g. *cosine* similarity) [7]. However, word embeddings trained by a mono-lingual corpus cannot be directly adapted to a multi-language classification task.

Multi-Lingual Embeddings: A word embedding is trained on a corpus of documents, normally in a single language. However, some recent works have examined training word embeddings for multi-lingual settings [8]–[10]. These approaches typically involve training a multi-lingual word embedding to capture the vocabulary similarities between different languages. For example, Chandar et al. [8] proposed an autoencoder approach to train a cross-lingual word embedding using sentence-aligned corpora containing documents in two languages. Another approach is proposed to learn bi-lingual or multi-lingual word embeddings using matrix factorisation [20]. However, such approaches often require the learning of new word embeddings from large parallel corpora. Hence, it is a time-consuming task that relies on large and good quality parallel corpora that are often not available for social media data. Besides, such approaches cannot reuse existing mono-lingual word embedding models that are publicly available.

Linear Translation Matrix: Different from learning multi-lingual embeddings to bridge the language gap, Mikolov et al. [11] proposed to learn a linear translation matrix to map from the embedding space of one language to that of another. This is motivated by the observed correlations between languages in word embedding spaces. Based on a translation corpus containing word alignments, such an approach has been studied by Mikolov et al. [11] in a machine translation task to retrieve the translations of a given word in another language. Mikolov et al. [11] showed that a promising accuracy is achieved despite its simplicity based on the study between English, Spanish and Czech. Nevertheless, their work focuses on the use of a linear translation in a machine translation task by retrieving the relevant words in the embedding of another language. How to use the linear translation approach in multi-language classification tasks has not been studied. As such, in this paper we adapt this approach to a multi-language Twitter election classification task. Using our Venezuela and Philippines election datasets, we thoroughly study the generalization of classifiers together with the linear translation approach in comparison to another transfer learning approach.

III. Linear Translation of Word Embedding Vectors

By projecting vector representations of words in different languages to low-dimensional spaces, Mikolov et al. [11] observed that the word vectors of similar words in different languages are related by a linear relationship. For example, the English numbers "one" to "five" and their Spanish translations have similar geometric arrangements in a low-dimensional space. Therefore, Mikolov et al. [11] assumed that a linear relationship exists between word embeddings in different languages. We apply this approach to learn a linear mapping from the embedding space of one language to that of another.

Suppose that $\mathbf{x_i}$ is the n-dimensional vector representation of word i in the source language and $\mathbf{y_i}$ the n-dimensional vector representation of its translation in the target language. By preparing a set of such word pairs, the task is to learn a $n \times n$ linear translation matrix $\mathbf{W}$ by optimising the following

TABLE I

QUERY TERMS USED TO RETRIEVE TWEETS AND STATISTICS OF THE LABELLED ELECTION DATASETS.

Dataset	Keywords
Venezuela	7D,6DGanaChávez,AbajoALalzquierda,CuentaRegresiva El6DGanaChaveze,ElCambioEstaEnLaEsquina,eleccionesAN guachiman6d,laManitoNiDeVaina,MiQuerenciaEsVenezuela,SOSVzla pasoloquepase,YoDefiendoMiRevolucion,VenezuelaQuiere,victoriaPerfecta
Philippines	PHVote,Halalan2016,PiliPinas2016,VotePH2016,PiliPinasDebates2016 RoxasRobredo,IVotePH,Elections2016,PHVoteDuterte,Eleksyon2016 MIRIAM2016,Binay,Poe, philippineelection2016

(a) Keywords used with the Twitter API to collect tweets related to each election

Dataset	Query terms
Venezuela	eleccion,violencia,votar,pistola,armas,ametralladora,ataque electora,muerto,miedo,muerte,asesinato,disparar,fraude muere,delincuente,herido,agreden,asesinar,guachiman,protesta
Philippines	violence,attack,dead,fraud,assault,protest,intimidation,unrest,cheating gunshot,racial,die,kill,threat,vote buying,murder,corrupt,election terrorize,ambush,explosion,shoot,fire,harass,injure,burn,selling vote

(b) Query terms used on Terrier IR platform

Dataset	Replacement	# Words	Election	Non-Election	Total
Venezuela (Spanish)	`NoRepl`	9,904	2,274	3,474	5,747
	`Repl`	7,945			
Philippines (English)	`NoRepl`	10,229	1,755	2,408	4,163
	`Repl`	8,635			

(c) Statistics of the annotated election datasets

problem:

$$\min_{W} \sum_{i=1}^{n} ||\mathbf{W}\mathbf{x_i} - \mathbf{y_i}||^2, \quad (1)$$

which can be solved with least squares or gradient descent.

Given a vector representation $\mathbf{x_i}$ in the source language, the predicted vector representation $\mathbf{\hat{y}_i}$ in the target language is obtained by:

$$\mathbf{\hat{y}_i} = \mathbf{W}\mathbf{x_i} \quad (2)$$

In a machine translation task, Mikolov et al. [11] used the predicted $\mathbf{\hat{y}_i}$ to retrieve its nearest neighbour in the embedding space of the target language as the translation of $\mathbf{x_i}$.

However, in this paper, we use the predicted vector representation $\mathbf{\hat{y}_i}$ directly without retrieving its nearest neighbour. Therefore, a linear translation matrix $\mathbf{W}$ is used to transform the vector representations between the English and Spanish word embeddings. For example, in our election classification task, when we apply a classifier learned from a Spanish dataset to an English dataset, we firstly transform the vector representations of English words into the Spanish word embedding space. Secondly, we apply the transformed representations within the learned classifier. Since the learned translation matrix $\mathbf{W}$ varies depending on the provided word pairs (i.e. a translation corpus), this may affect the classification performances of the CNN and SVM classifiers in our task and thus we discuss the impact of the translation corpus further with experimental results.

IV. EXPERIMENTAL SETUP

Our experiments are structured around the general problem of transferability using the translation matrix. In particular, we study how learned classifiers (e.g. SVM and CNN) transfer between our Venezuela and Philippines election datasets, which are in different languages and correspond to different election types. Moreover, we study how the translation corpus affects the performance of the linear translation approach in our task. By means of a number of experiments, we compare the performance of the linear translation approach to another classical transfer learning baseline. The remainder of this section details our pre-trained word embeddings (Section IV-A), the Venezuela and Philippines election datasets (Section IV-B), the translation corpus used to learn the translation matrix $\mathbf{W}$ (Section IV-C), the baselines (Section IV-D) and evaluation metrics (Section IV-E).

A. Word embeddings

We train English word embeddings and Spanish word embeddings from an English Twitter background corpus and a Spanish Twitter background corpus respectively. We choose Twitter corpus since when the type of corpus aligns with the dataset, the trained word embedding model can have a better word coverage on the dataset [16]. Both Twitter corpora are randomly collected from the period of 01/11/2015 to 31/12/2015 using the Twitter API. Over 20M English tweets are observed in the English Twitter corpus, while there are only over 5M Spanish tweets in the Spanish Twitter corpus. We apply stop-word removal and the Snowball stemmer for the corresponding language to all tweets. There exists different approaches to train word embedding models, such as Word2Vec [7], GloVe [21] and the approach proposed by Collobert et al. [18]. However, the trained models from these approaches are similar in terms of representing the word relationships. Therefore, we only train word embedding models using *Word2Vec*[1]. After training, there are $608K+$ unique words in the English word embedding model, while $196K+$ unique words occur in the Spanish word embedding

[1] the implementation in *deeplearning4j*: https://deeplearning4j.org/

TABLE II
STATISTICS OF THE SPLIT DATASETS USED IN TRAINING AND EVALUATION.

	Source D_s		Source D_s^v		Target D_t^o		Target D_t^u	
	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg
Venezuela ⇒ Philippines	1,441	2,236	833	1,237	1,579	2,167	176	241
Philippines ⇒ Venezuela	1,135	1,529	620	879	2,046	3,126	228	347

model. For the *Word2Vec* parameters, a context window size $W = 3$ and a dimension size $D = 500$ are used, since such settings already have exhibited good performances in a similar classification task [16]. For a word not appearing in a word embedding, also known as out-of-vocabulary (*OOV*), we simply initialise each dimension of their vector representations with *zero*. Therefore, all *OOV* words are treated as one word in the embedding space. We do not randomly initialise the vectors for *OOV* [12] because the randomness introduces noise when mapping such *OOV* words to another embedding space.

B. Datasets

In order to collect Twitter posts that are topically related to the Venezuela and Philippines elections, we use the Twitter API to collect posts that contain election-related keywords/hashtags (shown in Table I(a)) within the period of one month before and after the election dates. Over 7 million tweets are collected for the Venezuela election, while over 1.8 million tweets are collected for the Philippines election. Next, to permit human assessors to identify relevant (election-related) tweets without having to judge millions of tweets, we adopt a TREC-style pooling methodology [22]. In particular, we allow assessors to suggest queries, in response to which an IR system ranks the tweets each day, and k top-ranked tweets are added to the *pool* of tweets to be assessed; Table I(b) shows the query terms used. When ranking tweets, we use the Terrier IR platform [23] and the DFReeKLIM [24] weighting model that is designed for microblog retrieval. We select only the top $k = 7$ ranked tweets per query term per day, because this gives a tweet collection with reasonable size for our human annotators. Since the query terms for an individual election are picked by investigating related Twitter posts and newswire reports, they vary from one election to another election. At the final stage, the sampled tweets are labelled as: "Election-related" or "Not Election-related" by 5 experts in politics. Furthermore, for the Venezuela dataset, an agreement study was conducted on 482 randomly sampled tweets that were judged by all 5 assessors. We found a moderate agreement of 52% between all assessors using Cohen's *kappa*.

Following the aforementioned sampling and judging procedures, the Venezuela election dataset consists of 5,747 Spanish tweets, which cover the 2015 Venezuela Parliamentary Election. The second dataset covers the 2016 Philippines General Election, and consists of 4,163 English tweets. For consistency, we apply the same preprocessing, namely stop-word removal and stemming (as used in training the word embeddings) to our election datasets. As tweets often contain specific syntax such as Twitter handles and hashtags, they make a trained classifier specific to a particular event rather than generic to different events. To compare whether such special syntax affects the performance of the learned classifier, we build a variant of our datasets (denoted `Repl`) by replacing Twitter handles and hashtags with words "mention" and "hashtag". The original set of datasets is denoted as `NoRepl`. From the general statistics shown in Table I(c), we observe that both the Venezuela and Philippines election datasets are unbalanced. The positive class (Election related) is the minority class in both datasets. In `Repl`, the total number of unique words is reduced due to the replacement of hashtags and Twitter handles.

Using the Venezuela and Philippines election datasets, we consider two settings in this paper: "Venezuela (source domain in Spanish) ⇒ Philippines (target domain in English)" and "Philippines (source domain in English) ⇒ Venezuela (target domain in Spanish)". Following the training and evaluation strategy used by Pan et al. [25] in cross-domain text classification, we split our datasets into different subsets for each setting, as shown in Table II. We randomly sample 60% of instances from the source domain as D_s and the remaining 40% in the source domain as validation set D_s^v. 90% of instances from the target domain are sampled as the out-of-sample D_t^o that is used for evaluation; the remaining 10% in the target domain is the unlabelled subset D_t^u as required by the transfer component analysis (TCA) baseline.

C. Translation corpus

In order to learn the linear translation matrix $\mathbf{W}$, we generate two translation corpora that provide word-level alignment between Spanish words and English words. The two translation corpora are extracted from two different sources (our Twitter election datasets and Wikipedia) respectively and vary in size. Since the learned linear translation matrix $\mathbf{W}$ varies according to the provided translation corpora, we study whether the translation corpus affects the generalization of classifiers. Word pairs in the smaller corpus (denoted as `ELECT`) are only sampled from our Twitter election datasets and translated using the *Google translate* service. After the translation, we apply stemming to both the extracted words and their translations. Then, the duplicate translation pairs are removed. In particular, there are some stems shared by both English and Spanish words in our datasets. Such shared stems provide additional word-level alignments, which are helpful to the linear translation approach. The second corpus (denoted as `ELECT+Wiki`) covers both the `ELECT` and additional words sampled and translated from a Spanish Wikipedia snapshot dated 02/10/2015. We choose Wikipedia as the additional source to extract translation pairs since Wikipedia articles are better formatted compared to Twitter posts. By this means, we can extract more valid words from Wikipedia articles than that from Twitter posts. The Spanish Wikipedia dump contains $1M+$ documents and about $436K$ unique words. For training purposes, all the Spanish-English word pairs that appear in the translation corpus must exist in our corresponding word embeddings to avoid an *OOV* problem. After the same preprocessing is used in training the word embeddings, the `ELECT+Wiki` translation corpus consists of 9,410 Spanish-English word pairs, while the `ELECT` corpus only consists of 4,440 word pairs. The number of translation pairs covered by `ELECT+Wiki` is more than twice as that of the `ELECT` corpus. Hence, it allows us to study whether a larger translation corpus can benefit the generalization of classifiers in the multi-language Twitter classification task.

D. Baselines

In order to study the effectiveness of the linear translation, we compare it to the following baselines:

TCA: Transfer component analysis (TCA) is a dimensionality reduction-based transfer learning approach. In order to apply TCA to our dataset using word embeddings, we represent a tweet by averaging the word embedding vectors along each dimension for all the words in the tweet [26]. Such tweet representations allow us to run TCA without out-of-memory errors that would be caused by kernel learning in TCA. We use D_s and D_t^u to learn the transfer matrix $\mathbf{W}$, which is further applied to transform the out-of-sample subset D_t^o to the source domain. Following the settings studied by Pan et al. [25], we only preserve the first 30 dimensions of the transformed features. The transformed source dataset $\hat{D}_s$ is then used to train the classifiers (e.g. CNN and SVM), while the transformed out-of-sample dataset $\hat{D}_t^o$ is used for evaluation.

NoLT: As a basic baseline that does not use transfer learning (NoLT), we train the classifiers on D_s without using any transfer learning approach and test it directly on D_t^o.

Upper: As an upper bound on accuracy, we train the classifiers using the target domain D_t^u and test it also on the target domain D_t^o. Therefore, it gives the best performance we can obtain on a single target domain dataset.

Random: This baseline makes random predictions to D_t^o based on the class distribution of the training dataset D_s.

Majority: This baseline classifies all the instances in D_t^o as the negative class, which is the majority class in our datasets.

E. Training, Hyper-parameters & Metrics

To make the results comparable, we apply all the aforementioned approaches to the convolutional neural networks (CNN) and support vector machines (SVM) classifiers by keeping the settings for all the experiments. For the CNN classifiers, we follow the settings suggested by Kim [12], except that only one size filter $m = 1$ is used. This different setting for the filter size allows the classifier to capture significant features for each word vector, which also yielded a better performance in our initial experiments and compared to $m = \{3, 4, 5\}$ of Kim [12]. For the SVM classifier[2], we set parameter $c = 1$. For both CNN and SVM classifiers, tweets are converted into vector representations by concatenating the corresponding word embeddings vectors of all the words in a tweet. Due to the variable length of tweets, short tweets are padded to the length of the longest tweet using a special token as the *OOV* words. To evaluate all of the approaches, we run 5 repetitions for the same experiment due to the randomness in the CNN setup and report the average precision, recall, F1 score and balanced accuracy (i.e. the average accuracy on either class, denoted as BAC) over the 5 repetitions.

In the following, we address three research questions:

- **RQ1**: “By applying text preprocessing to the datasets as mentioned in Section IV-B, do the generalised datasets improve the effectiveness of the linear translation approach?”
- **RQ2**: “Is an election-specific translation corpus more effective than a larger translation corpus, when using the linear translation approach?”
- **RQ3**: “Is the linear translation approach more effective than other baseline approaches such as Transfer Component Analysis (TCA)?”.

We will address **RQ1** and **RQ2** in Section V-A and address **RQ3** in Section V-B.

V. Results

In this section, we first report our experimental results of using the linear translation approach on: (1) Two dataset variants, namely `NoRepl` and `Repl`; (2) Two translation corpora `ELECT` and `ELECT+Wiki` that cover different sizes of translation word pairs. Then we compare the linear translation approach to baselines listed in Section IV-D.

A. Effect of translation corpus and Twitter handles & hashtags in the linear translation

Compared to traditional text corpora such as Wikipedia, the specific syntax of tweets (e.g. handles and hashtags) often makes a trained classifier sensitive to a certain topic (i.e. a hashtag is a strong indicator of a specific topic), and therefore could affect the generalization of the classifier. To evaluate the impact of the Twitter handles and hashtags in cross-election classification, we compare the results of the two dataset variants, namely `NoRepl` and `Repl`, which are introduced in Section IV-B. In addition, we also study the effect of the size of translation corpus. Using various translation matrices $\mathbf{W}$ that are learned from two translation corpora `ELECT` and `ELECT+Wiki`, we translate the word representations from the embedding space of one language to that of another. For example, in Venezuela ⇒ Philippines (denoted as *V* ⇒ *P*), we aim to train a classifier on the Venezuela election dataset (in Spanish) and test the classifier on the Philippines election dataset (in English). Since the classifier is trained using the Spanish word embeddings, we have to map our English words in the Philippines dataset to the Spanish word embeddings space using Eq. (2). In this way, we firstly transform the vector representations of the English tweets in the Philippines dataset and then apply the classifier to classify the transformed instances. The classification results of using various translation corpora are shown in Table III, where the first two columns show the cross-election tasks and the datasets used. By varying the translation corpus, we report three metrics (precision, recall

[2] the *LinearSVC* model in *scikit-learn* is used: http://scikit-learn.org

TABLE III
CLASSIFICATION RESULTS OF USING VARIOUS TRANSLATION CORPORA. BEST RESULTS ARE HIGHLIGHTED IN BOLD. † INDICATES THE RESULT IS STATISTICALLY SIGNIFICANT COMPARED TO THE CORRESPONDING NOREPL USING THE SAME TRANSLATION CORPUS.

Task	Replacement	Translation corpus	Precision (%)	Recall (%)	F1 Score (%)
Venezuela ⇒ Philippines (V ⇒ P)	NoRepl	ELECT	48.7	46.9	46.7
		ELECT+Wiki	48.7	42.4	43.5
	Repl	†ELECT	**50.3**	58.6	**54.1**
		†ELECT+Wiki	49.1	**60.6**	54.0
Philippines ⇒ Venezuela (P ⇒ V)	NoRepl	ELECT	57.4	51.4	54.2
		ELECT+Wiki	**57.9**	49.6	53.2
	Repl	†ELECT	55.9	**60.8**	**58.2**
		†ELECT+Wiki	54.7	56.4	55.5

(a) Results of CNN classifiers

Task	Replacement	Translation corpus	Precision (%)	Recall (%)	F1 Score (%)
Venezuela ⇒ Philippines (V ⇒ P)	NoRepl	ELECT	56.8	28.8	38.2
		ELECT+Wiki	52.0	20.6	29.5
	Repl	ELECT	**57.0**	**36.3**	**44.3**
		†ELECT+Wiki	55.4	31.9	40.5
Philippines ⇒ Venezuela (P ⇒ V)	NoRepl	ELECT	52.4	55.5	54.0
		ELECT+Wiki	**54.9**	48.3	51.4
	Repl	†ELECT	49.8	**67.6**	**57.4**
		†ELECT+Wiki	49.0	66.0	56.2

(b) Results of SVM classifiers

and F1) for the classification performances using the dataset variants NoRepl and Repl.

As shown in Table III, we list classification results of both CNN classifiers (Table III(a)) and SVM classifiers (Table III(b)). For both tasks of $V \Rightarrow P$ and $P \Rightarrow V$, the classification performances when using the Repl dataset outperform those achieved with the NoRepl dataset. In particular, by conducting *McNemar's* test [27], the CNN classifiers using the dataset Repl and translation corpus ELECT achieved significant improvements for $V \Rightarrow P$ and $P \Rightarrow V$ compared to the classifiers using dataset NoRepl and translation corpus ELECT. For the SVM classifiers, the use of Repl and ELECT also achieved a statistically significant improvement for the task $P \Rightarrow V$ compared to the SVM classifiers using NoRepl and ELECT. In short, this means that the performance difference between Repl and NoRepl is considered to be statistically significant for both tasks, and hence, for **RQ1**, we find that text preprocessing does result in increased effectiveness when using the linear translation approach.

Furthermore, we show that this approach can be adapted to different classifiers. In Table III, a similar trend is observed that the replacement Repl improves recall and yields better F1 scores for both the SVM and CNN classifiers. For example, Table III(a) shows that for the CNN classifiers, recall and F1 are improved for the task $V \Rightarrow P$ by 16% when the translation corpus ELECT is used. In the $P \Rightarrow V$ task, although there is a slight drop in precision when Repl and ELECT are used, the recall and F1 scores are improved by 18% and 7%, respectively. The result shows that, by replacing the Twitter handles and hashtags with the general words "mention" and "hashtag" respectively, we can generalise the representations of tweets for the trained CNN and SVM classifiers. Moreover, when using Repl, the performances of the classifiers are more balanced in terms of precision and recall between the two tasks.

When the results are compared between ELECT and ELECT+Wiki within the same variant of datasets, we observe that the two translation corpora have similar performances for both CNN and SVM classifiers. In most cases, ELECT only has a slightly better performance in the recall and F1 scores when replacement Repl is used. Although the translation corpus ELECT+Wiki covers additional translation pairs extracted from Wikipedia, the highlighted results show that in our task a larger translation corpus does not always improve the generalization of the CNN and SVM classifiers. Therefore, to answer **RQ2**, we conclude that an election-specific translation corpus is as effective as a larger translation corpus. Thus, additional translation pairs that are extracted from Wikipedia articles do not yield a translation matrix $\mathbf{W}$ that gives better classification results on the target domain dataset.

From Table III, by comparing the highlighted results between the CNN and SVM classifiers, the CNN classifiers have a more balanced performance for both tasks of $V \Rightarrow P$ and $P \Rightarrow V$. In particular, the SVM classifiers cannot achieve comparable recall and F1 for the task $V \Rightarrow P$ in contrast to the task $P \Rightarrow V$. Therefore, in the next experiments where the transferability is studied, we only use the CNN classifiers.

B. Transferability

To evaluate the transferability of the linear translation approach (denoted CNN+LT), we compare it to all of our baselines in a multi-language Twitter election classification task, including a classical transfer learning approach, namely transfer component analysis (TCA). The obtained results are shown in Table IV where the first column shows the tasks, and the second column shows the classifiers we trained by using different approaches. We report four metrics, namely precision, recall, F1 score and balanced accuracy (denoted BAC) in the last four columns. For the CNN classifiers trained using TCA, we further study whether they can benefit from the generalised dataset Repl. However, since Random, Majority, Upper and CNN+NoLT do not adapt any transfer learning approach, we do not apply Repl to these baselines.

TABLE IV
Classification results of linear translation and baselines. The best results in each setting are highlighted in bold without considering the Upper baselines. † indicates the result is statistically significant compared to Random, Majority, CNN+NoLT and CNN+TCA baselines.

Task	Classifier	Precision (%)	Recall (%)	F1 Score (%)	BAC (%)
	Random	41.7	37.9	39.8	49.7
Venezuela	Majority	0.0	0.0	0.0	50.0
⇓	CNN+NoLT	**74.3**	1.7	3.4	50.6
Philippines	CNN+TCA (NoRepl)	65.6	32.2	43.1	**59.9**
	CNN+TCA (Repl)	57.3	11.9	19.7	52.8
(V ⇒ P)	CNN+LT (NoRepl)	48.7	46.9	46.7	54.4
	†CNN+LT (Repl)	50.3	**58.6**	**54.1**	58.2
	CNN+Upper	80.2	71.9	75.8	79.4
	Random	39.9	41.8	40.9	50.3
Philippines	Majority	0.0	0.0	0.0	50.0
⇓	CNN+NoLT	46.0	3.6	6.0	50.5
Venezuela	CNN+TCA (NoRepl)	49.1	25.5	32.5	53.6
	CNN+TCA (Repl)	34.1	19.2	23.7	47.9
(P ⇒ V)	CNN+LT (NoRepl)	**57.4**	51.4	54.2	63.2
	†CNN+LT (Repl)	55.9	**60.8**	**58.2**	**64.7**
	CNN+Upper	79.8	69.6	74.4	79.1

As shown in Table IV, the CNN+LT classifier trained using the linear translation indeed significantly outperforms all of other baselines except the Upper baseline in terms of recall and F1 measures. This result positively answers **RQ3**. We note that the precision, recall and F1 of the Majority baselines are zeros because the majority class in both election datasets is the negative class (i.e. Not Election-related). In this case, none of the "Election-related" instances are correctly classified by the Majority baselines, which yields zero score in the three metrics. CNN+NoLT is much worse than the random classifier in both recall and F1 scores. In particular, CNN+NoLT shows high precision but low recall, which is similar to the Majority baseline that is in favour of predicting tweets in another language as "Not Election-related". In other words, the CNN classifier of CNN+NoLT fails to identify "Election-related" tweets, and therefore it cannot transfer knowledge from one election in language A to another election in language B without applying the linear translation.

However, TCA can benefit from the similarities between word embeddings, and thus finds the transfer components of different word embeddings in the kernel space. The results of CNN+TCA indeed show that TCA helps the CNN classifiers to improve the balanced accuracy, and thus yields better transferability than CNN+NoLT. Nevertheless, TCA does not benefit from the generalised dataset Repl that has been shown earlier to be useful to the linear translation approach. In both *V ⇒ P* and vice versa, CNN+TCA (Repl) is less effective than CNN+TCA (NoRepl). This shows the difference between TCA and the linear translation in transferring the word embedding features. TCA aims to capture the most k common components among word embedding features, while the linear translation learns a linear mapping to bridge two word embeddings. Therefore, by adding the generalised terms "mention" and "hashtag" in the translation corpus, the generalised terms in different word embeddings can be easily related by linear translation. Therefore, the constructed vector representations of tweets are similar to the target language when the linear translation is used. However, such a setting does not help TCA since it learns transfer components on the tweet-level vector representations rather than using individual word vectors.

When we compare CNN+LT to the CNN+Upper that achieved the highest attainable performance by training on the target domain, we note that the upper baselines have significantly better performance than CNN+LT, especially on precision. However, CNN+LT has already achieved significantly better performance compared to the other baselines using *McNemar's* test, particularly on recall, which shows the potential of the linear translation approach in our task. In summary, we show that, without using any training instances from the target election dataset, linear translation is a simple yet significantly effective way to transfer word embedding features between different domains in election datasets.

C. Discussion

In Table V, we list some examples in our election dataset to study the difficulties and possible future directions in transfer learning for the multi-language Twitter election classification task. From Table V, we note the difficulty of this task, due to various problems, for example: (1) The content of the tweets (e.g. *P1*, *P3* and *V3*) contain no clear election-related words except some hashtags, Twitter handles and names of politicians. In such examples, we note that the hashtags and politicians' names are salient indicators of election-related tweets such as "#PHVoteRoxas", "PPCRV" and "Tintori"; (2) In the Venezuela dataset, many of the election-related tweets involve reporting violent events such as *V2*. This contrasts with the Philippines dataset, where many tweets were reporting violent events about non-election-related military conflicts such as *P2*. To address both problems, a better approach is required to more comprehensively identify hashtags and Twitter handles relevant to elections. Therefore, such indicators can then be re-used to improve the transferability and both the precision and recall scores. For instance, it may be possible to better generalise election-related terms such as the politicians' names and party names. Accordingly, "election-related" and "not election-related" tweets can be better distinguished through the presence of such election-related terms.

TABLE V

EXAMPLES OF WRONG PREDICTIONS USING LINEAR TRANSLATION. "+" ("−") DENOTES ELECTION-RELATED (NOT ELECTION-RELATED).

			Philippines Election
No.	Label	Prediction	Text
P1	+	−	#PHVoteRoxas: "Anyone who laughs at the ultimate assault on the dignity of women should not be allowed to wield power."
P2	−	+	3 Philippine soldiers killed, 2 injured in clash with communist guerrillas Saturday on central island...
P3	+	−	PPCRV applies cold water to burned area #LeniIsMyVP
			Venezuela Election
No.	Label	Prediction	Text
V1	−	+	translated: #Somos112 Venezuela does not want more violence
V2	+	−	translated: Armed groups attacked in Guarico VP activists as they waited Tintori
V3	+	−	translated: Only #22Dias for change, for the birth of the new Venezuela #6Dic

VI. CONCLUSIONS

In this paper, we adapt the linear translation approach to a multi-language classification task on Twitter election datasets. To the best of our knowledge, this is the first application of linear translation for multi-language classification. By learning a translation matrix $\mathbf{W}$ using a translation corpus, we bridge the feature gap between the Venezuela (in Spanish) and the Philippines (in English) election datasets. In particular, applying text preprocessing (i.e. replacing the Twitter handles and hashtags with the words "mention" and "hashtag") leads to more generalisable classifiers (i.e. CNN and SVM) across our two datasets. We also note that a smaller election-specific translation corpus has a similar effectiveness as a much larger translation corpus. Finally, compared to other baseline approaches including the transfer component analysis (TCA), our results show that without using any training data from the target language, the linear translation approach has better transferability.

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] A. Bermingham and A. F. Smeaton, "On using Twitter to monitor political sentiment and predict election results," in *Proc. of SAAIP workshop at IJCNLP*, 2011.

[2] A. Fang, I. Ounis, P. Habel, C. Macdonald, and N. Limsopatham, "Topic-centric classification of Twitter user's political orientation," in *Proc. of ACM SIGIR*, 2015, pp. 791–794.

[3] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series." in *Proc. of ICWSM*, vol. 11, no. 122-129, 2010, pp. 1–2.

[4] J. Ratkiewicz, M. Conover, M. R. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, "Detecting and tracking political abuse in social media." in *Proc. of ICWSM*, vol. 11, 2011, pp. 297–304.

[5] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. of ICML*, 2007, pp. 193–200.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[8] S. Chandar, S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V. C. Raykar, and A. Saha, "An autoencoder approach to learning bilingual word representations," in *Proc. of NIPS*, 2014, pp. 1853–1861.

[9] S. Eger and A. Hoenen, "Language classification from bilingual word embedding graphs," *arXiv preprint arXiv:1607.05014*, 2016.

[10] H. Zhou, L. Chen, F. Shi, and D. Huang, "Learning bilingual sentiment word embeddings for cross-language sentiment classification," in *Proc. of ACL*, 2015, pp. 134–141.

[11] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv:1309.4168*, 2013.

[12] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. of EMNLP*, 2014, pp. 1746–1751.

[13] A. Severyn, M. Nicosia, G. Barlacchi, and A. Moschitti, "Distributional neural networks for automatic resolution of crossword puzzles," in *Proc. of IJCNLP*, 2015.

[14] A. Severyn and A. Moschitti, "UNITN: Training deep convolutional neural network for Twitter sentiment classification," in *Proc. of SemEval*, 2015, pp. 464–469.

[15] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proc. of ACL*, 2014, pp. 1555–1565.

[16] X. Yang, C. Macdonald, and I. Ounis, "Using word embeddings in Twitter election classification," in *Proc. of Neu-IR workshop at SIGIR*, 2016.

[17] S. Ebert, N. T. Vu, and H. Schütze, "CIS-positive: Combining convolutional neural networks and SVMs for sentiment analysis in Twitter," in *Proc. of SemEval*, 2015, p. 527.

[18] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. of ICML*, 2008, pp. 160–167.

[19] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, pp. 1137–1155, 2003.

[20] T. Shi, Z. Liu, Y. Liu, and M. Sun, "Learning cross-lingual word embeddings via matrix co-factorization," in *Proc. of ACL*, 2015, pp. 567–572.

[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. of EMNLP*, 2014, pp. 1532–1543.

[22] E. M. Voorhees and D. K. Harman, *TREC: Experiment and Evaluation in IR*. MIT Press, 2005.

[23] C. Macdonald, R. McCreadie, R. L. Santos, and I. Ounis, "From puppy to maturity: Experiences in developing Terrier," in *Proc. of OSIR workshop at SIGIR*, 2012.

[24] G. Amati, G. Amodeo, M. Bianchi, G. Marcone, F. U. Bordoni, C. Gaibisso, G. Gambosi, A. Celi, C. Di Nicola, and M. Flammini, "FUB, IASI-CNR, UNIVAQ at TREC 2011 microblog track." in *Proc. of TREC*, 2011.

[25] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[26] P. Wang, J. Xu, B. Xu, C.-L. Liu, H. Zhang, F. Wang, and H. Hao, "Semantic clustering and convolutional neural network for short text categorization," in *Proc. of IJCNLP*, vol. 2, 2015, pp. 352–357.

[27] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.