INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Bilingual Word Embeddings for Cross-Lingual Personality Recognition Using Convolutional Neural Nets

Farhad Bin Siddique, Pascale Fung

Human Language Technology Center
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
fsiddique@connect.ust.hk, pascale@ece.ust.hk

Abstract

We propose a multilingual personality classifier that uses text data from social media and Youtube Vlog transcriptions, and maps them into Big Five personality traits using a Convolutional Neural Network (CNN). We first train unsupervised bilingual word embeddings from an English-Chinese parallel corpus, and use these trained word representations as input to our CNN. This enables our model to yield relatively high cross-lingual and multilingual performance on Chinese texts, after training on the English dataset for example. We also train monolingual Chinese embeddings from a large Chinese text corpus and then train our CNN model on a Chinese dataset consisting of conversational dialogue labeled with personality. We achieve an average F-score of 66.1 in our multilingual task compared to 63.3 F-score in cross-lingual, and 63.2 F-score in the monolingual performance.

Index Terms: Computational Personality Recognition, Big Five, Multilingual Analysis, Convolutional Neural Nets

1. Introduction

Computational personality recognition has become increasingly important in a wide variety of applications, including personalised virtual agents, recommendation systems, and people assessment for job suitability. Personality is a significant factor of communication in human-human interaction [1, 2], and as virtual agents and dialogue systems get more intelligent, we expect them to identify and adapt to different user personalities. Having such a personality recognition module in dialogue systems will enable us to have more intelligent and personal human-agent conversations in the future [3, 4].

The Big Five personality model [5] has been the most formally recognized by psychologists and has been used to quantify user personality in terms of scores across five different dimensions:

- Extraversion vs Introverted
- Conscientiousness vs Careless
- Agreeableness vs Detached
- Neuroticism vs Emotionally Stable
- Openness to Experience vs Cautious

Traditionally, scores across these five dimensions are measured via self-assessment where users fill up a form answering various questions on a scale and the answers are mapped to scores in the Big Five, most commonly used is the NEO Personality Inventory [6]. This sort of assessment is not suitable for many real-time applications and therefore we have developed the need for automatic personality assessment.

Most of the past work to identify user personality from text involves lexical feature extraction to train Support Vector Machines (SVMs) and other non-linear classifiers. The feature engineering involved is heavily dependent on the dataset used, and the features cannot be shared across multiple languages. There is inadequate work done in the task of language independent personality classification. This maybe due to limited personality labeled data available in languages other than English. For example, there is not enough work done in identifying personality from languages such as Chinese [7]. Therefore we propose a cross-lingual model that can achieve the task of assessing personality in both the source and target language by using trained bilingual word embeddings, which enables the model to perform well when tested on the target language. Such a method tries to alleviate the problem of data scarcity in low resource languages.

Researchers have always questioned whether personality assessment can be applied across languages. Can a recognition model trained in language A be applied to language B in some fashion? We set out to find out. We design two types of models - (1) a cross-lingual model that has been trained in language A is used to recognize personality traits in language B; (2) a multilingual model where language A and B are combined to train the recognizer which is then used to recognize personality traits in language B. We would like to compare these two models to a monolingual model trained in language B.

2. Related Work

Automatic assessment of personality from text started as early as 2006 [8]. [8] classified blog authors based on Big Five using n-gram as features and Naive Bayes (NB) as the learning algorithm. [9] used two different kinds of datasets, both conversational and written texts along with two different sources of lexical features, LIWC [10] and MRC [11], and classified personality scores and classes using SVMs and M5 trees. More recently, social media texts have been used in classifying user personality. [12] predicted personality of 279 Facebook users using both linguistic (LIWC) and social features (number of friends, likes). The Workshop on Computational Personality Recognition (WCPR) [13] invited papers on this very task and released personality labeled datasets including the Facebook and Youtube Vlogs datasets that are used for training our model described in this paper. [14] used 2000 frequent trigram features and trained a SVM classifier. [15] used LIWC features and tested three different learning algorithms, NB, SVMs and

http://dx.doi.org/10.21437/Interspeech.2017-1379

nearest neighbors (kNNs). Although some of these papers report decent results, as mentioned before, selecting the features to use depending on the dataset makes the model not so robust, and therefore it is not possible to train a language independent model, or extend the model to outside-domain corpora.

Deep learning models have already shown groundbreaking results in fields such as speech recognition [16] and computer vision [17]. In natural language field, work has been done in representing language in higher dimensional vectors that can contain valuable semantic information. Deep neural models have been trained to learn such context relevant vectors from large text corpus [18, 19]. The idea is to obtain vectors, called word embeddings, that will maintain a low cosine distance across semantically similar words. Convolutional Neural Nets can carry out convolution operations on the input layer to extract local features [20], and such a model is used on top of the learned word embeddings to solve text classification tasks [21, 22].

Learning word alignments using word context information from both parallel and non-parallel corpora has been done since the early 90s [23, 24]. Cross-lingual word embedding algorithms try to represent the vocabularies of two or more languages in one common continuous vector space. Bilingual word embeddings are trained on two languages where the learned vectors are expected to hold contextual information across both languages. This enables a model to overcome the problem of sparsity in the amount of data available in multiple languages for different classification tasks [25].

3. Methodology

We propose a method to train unsupervised bilingual embeddings from a parallel corpus. These embeddings are then used to represent the text data in the higher dimensional representation, where each word is represented by its corresponding learned embedding. This input is fed into a one layer CNN model that extracts features from the text using the filters assigned and maps the features to a softmax function output, which is essentially a probability estimation of the binary classification of each personality trait.

3.1. Bilingual Word Embedding

There are several different approaches described in the literature to train cross-lingual embeddings from parallel corpora, some using document level alignments [26], while others requiring sentence or word level alignments [27, 28]. We have chosen and implemented the bilingual skip gram (Bi-skip gram) model proposed by [29]. It is an extension of the original monolingual skip-gram model [19]. The Bi-skip gram model tries to minimize the cost function across two languages when taking into account the word context information of each word in both the languages. The simplified model architecture is shown in Figure 1.

The objective function to minimize includes the bilingual part along with the monolingual context:

$$(\mathbf{X}^*, \mathbf{Y}^*) \leftarrow \arg\min_{\mathbf{X},\mathbf{Y}}$$

$$\alpha\big[Mono_1(\mathbf{X}) + Mono_2(\mathbf{Y})\big] + \beta Bi(\mathbf{X}, \mathbf{Y}) \quad (1)$$

where, $(\mathbf{X}^*, \mathbf{Y}^*)$ are the final word vectors of languages L_1 and L_2 respectively, learned from the context after being initialized to some constant vectors. The monolingual part, similar to the skip gram model, tries to predict the context words in the same language given a certain word. The bilingual part deals with predicting the context y_c in language L_2, given a word x in language L_1, and vice versa. The parameters α, β determine how we want to weight our model's bilingual and monolingual performance. The objective functions can be expressed as follows:

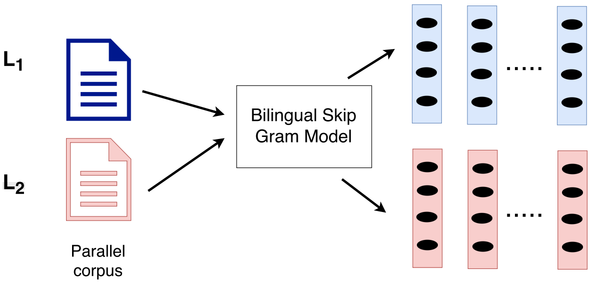

Figure 1: *Bilingual word embedding training to produce the word vectors, using parallel corpus of language L_1 and L_2*

$$Mono_1(\mathbf{X}) = -\sum_{x\in X}\sum_{x_c\in T_1(x)} \log P(x_c|x) \quad (2)$$

$$Mono_2(\mathbf{Y}) = -\sum_{y\in Y}\sum_{y_c\in T_2(y)} \log P(y_c|y) \quad (3)$$

$$Bi(\mathbf{X}, \mathbf{Y}) = -\sum_{(x,y)\in Z}\sum_{x_c\in T_1(w)} \log P(x_c|y)$$

$$-\sum_{(x,y)\in Z}\sum_{y_c\in T_2(y)} \log P(y_c|x) \quad (4)$$

where, $x \in X$ and $y \in Y$ are the words present in the vocabulary of each language, $T_1(x)$ is the context of x in language L_1, $T_2(y)$ is the context of y in language L_2, and Z is the set of words present in the contextual window size. As we can see, the model is similar to training four different skip gram models where each identifies a different combination of word context pairs. Since this model takes into account the semantic information of words in both the monolingual and bilingual context, it does not deteriorate the monolingual performance while training the bilingual word embeddings.

3.2. Convolutional Neural Network

Convolutional Neural Nets usually contain a CNN layer that performs convolution operations across the input matrix space in order to get local features. The convolutional filters used are trained in the model. CNNs have gained popularity recently in efficiently carrying out the task of text classification [21, 22]. In most such tasks, pre-trained word vector representations like the Google word2vec [19] are used to represent the text in the input layer. We propose a model that takes as input, text represented with our pre-trained bilingual word embeddings, followed by a single CNN layer, and then a max pooling layer. At the end we have a fully connected layer with non-linear activation and softmax to map the features to the binary classification task for each personality trait.

In our input layer, we have each text document to be classified, represented as a matrix, with each word represented by its respective bilingual embedding. Depending on the convolutional window size, the convolution operation slides along the

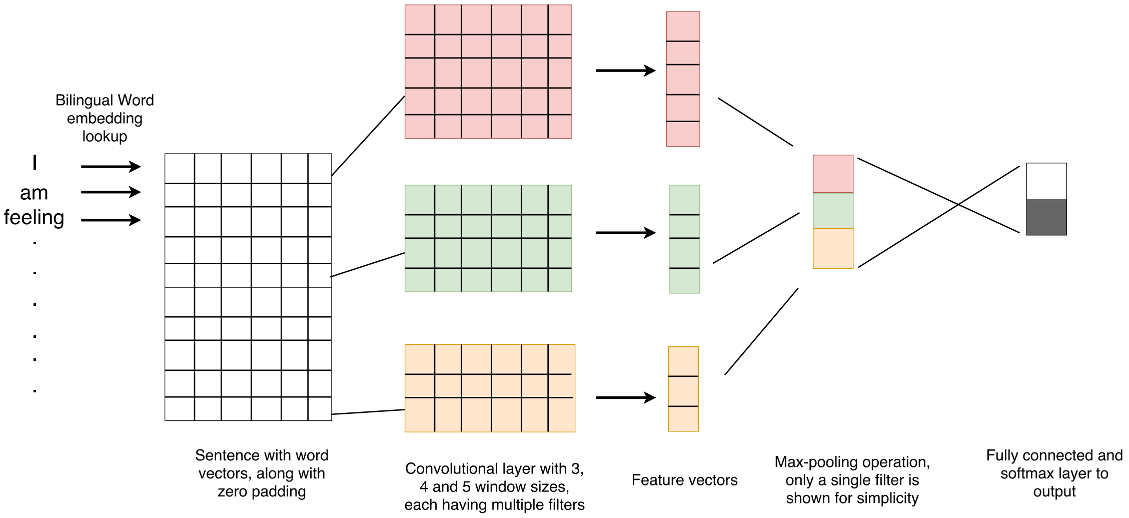
Bilingual Word embedding lookup
I
am
feeling
Sentence with word vectors, along with zero padding
Convolutional layer with 3, 4 and 5 window sizes, each having multiple filters
Feature vectors
Max-pooling operation, only a single filter is shown for simplicity
Fully connected and softmax layer to output

Figure 2: *CNN model for binary personality classification*

matrix to get feature vectors. We use 3, 4 and 5 to be our window sizes, and these typically represent 3, 4 and 5-gram features extracted from the text. Multiple filters are used for each window size, and the maximum features are chosen in the max-pooling layer. The final feature vector is then mapped to the binary classification via a fully connected layer and softmax function. The architecture is given in Figure 2.

4. Experimental setup

4.1. Corpora

For the bilingual word embedding training, we use a Chinese-English parallel corpus consisting of 2.2 Million sentences collected from eight different domains, called the UM corpus [30]. This dataset was created by extracting information from web-crawled data, and high quality parallel text sources were used. The data was crawled from a wide variety of domains in order to increase the varieties of content. The corpus is sentence-aligned, and we use the Jieba segmenter[1] to pre-process the Chinese sentences and tokenize them into words. A segmenter is needed since Chinese texts are delimited by sentences and not words, unlike English and other languages. The Chinese data contains around 50 Million words in total and the vocabulary size is 160,473, while the English data has around 48.5 Million words in total with a vocabulary size of 160,262.

Two personality labeled datasets were used for our task, collected by the myPersonality[2] project. The Facebook dataset [31] consists of around 9,900 status updates from 250 users labeled with their Big Five personality traits, which was determined by self assessment forms the users filled up. The Youtube dataset [32] consists of transcriptions of 404 vloggers labeled with their perceived personality scores collected via mechanical turk. These datasets were released for the *Workshop on Computational Personality Recognition* (WCPR) [33]. The Facebook and Youtube personality datasets were combined and used for training. A median split of the Big Five scores is done to divide each of the big five personality groups into two classes, and so the task was five different binary classification tasks, one for each trait.

We also use a Chinese personality labeled dataset called the BIT Speaker Personality Corpus [34] produced by the Beijing Institute of Technology. This dataset contains 498 Chinese speech clips, each of length around 9-13 seconds and each labeled with Big Five scores by five judges in total. We take the average score of the five judges, and then do a median split of scores for each personality trait. We use an automatic speech recognition (ASR) system to get the text transcription results of each audio file, and then use the Jieba segmenter to tokenize the Chinese text into words.

4.2. Experiments

We carry out different train and test experiments with the two language datasets we have, to compare our model's monolingual, cross-lingual and multilingual performance. For each experiment, we train five different binary classifiers, one for each personality trait. Each of them is tested on the same Chinese test set containing 98 different transcriptions of audio clips, labeled with personality. The experiments are described below:

- **Monolingual**: For comparison, we train monolingual word embeddings for Chinese, using the original skip-gram model [19]. For the training, we use Chinese text data from Chinese articles collected from the *Chinese Wikipedia Dump*[3]. After pre-processing to tokenize the Chinese text using Jieba segmenter, the dataset contains around 207 million words in total and has a vocabulary size of 460,882. We use the trained monolingual embeddings to represent the Chinese text at the input layer, and among the 498 different transcriptions from the audio clips, we use 400 for training and test our model on the test set of 98.
- **Cross-lingual**: We train our CNN model on the English dataset with the bilingual embeddings as input to represent the text. We then test our model on the Chinese test set containing 98 Chinese clips with transcriptions labeled with personality.
- **Multilingual**: We train our model on a combined dataset of English and Chinese using our pre-trained bilingual

[1] https://github.com/fxsjy/jieba

[2] http://mypersonality.org/

[3] https://dumps.wikimedia.org/zhwiki/

Table 1: *F-score results for the monolingual, cross-lingual and multilingual performance of the CNN model compared to the SVM baseline, all tested on the same Chinese test set. The highest performance for each trait is given in bold.*

	Extraversion	**Conscientiousness**	**Agreeableness**	**Neuroticism**	**Openness**	*Average*
Monolingual	61.8	62.1	**65.2**	60.2	66.7	63.2
Cross-lingual	65.1	63.2	58.6	65.7	64.1	63.3
Multilingual	**66.7**	**64.7**	63.8	**66.2**	**68.9**	**66.1**
Baseline SVM	59.3	56.3	58.9	57.7	59.2	58.3

word embeddings to represent the text data as input to our CNN. We use all 654 users' English text data plus 400 of the Chinese transcriptions, for training, and test our model on the remaining 98 Chinese test set.

- **Baseline SVM**: In order to compare our model's results with a feature based implementation, we train a baseline SVM classifier, using Linguistic Inquiry and Word Count (LIWC) [10] dictionary to extract different lexical features. The features include emotion features such as positive and negative emotion words, and other qualitative features, like number of pronouns, verbs, conjunctions, and so on. We use the Chinese LIWC dictionary for extracting the features. We train the SVM on 400 Chinese transcriptions and test the model on the test set of size 98.

4.3. Hyper-parameters and regularization

For the bilingual word embeddings, we choose the parameters α and β both to be 1, in order to give the same weightage to the mono and bi part of our training. We use negative sampling of the words, to prevent the estimation of computationally inefficient normalization terms. Negative sampling tries to differentiate the data from noise via logistic regression [35]. We train 100 dimensional vectors, and choose our context window size to be 10, and use 30 as the number of negative samples in training, and we run the model for 10 iterations.

For the monolingual word embeddings for Chinese, we train the skip gram model with vectors of dimension 100, and choose our context window size to be 5, and use 10 negative samples, and train our model for 5 iterations.

The convolution window sizes of 3, 4 and 5 are used in order to represent n-gram features from the convolutional filters, and a total of 128 filters are trained in parallel for each window size. For regularization, we use the l2 loss regularization with lambda tuned from [0, 0.01, 0.1] for each model, and we use dropout [36] with a keeping probabibility tuned from [0.4, 0.5, 0.6] for each model. We used rectified linear (ReLu) to add non-linearity, and the Adam optimizer [37] was used for the training update at each step, with the learning rate set to be 10^{-4}.

For our SVM we use RBF kernel, and tune our gamma values from [0.1, 1, 10, 100] and choose the one yielding the highest performance for each classifier.

All the hyper-parameters tuning are done via cross validation by using a 10% split of the training set as the development set.

4.4. Results and Discussion

The F-score results of the different experiments of our model are given in Table 1. As we can see in the results, all three of our models beat the baseline feature based SVM performance for this dataset. Our cross-lingual experiment, when trained on English using the bilingual embeddings, and tested on Chinese, outperforms the monolingual performance in traits apart from Agreeableness and Openness to Experience. Also, our multilingual performance, when trained on both English and Chinese dataset using bilingual embeddings, and tested on the Chinese test set, performs the best at every trait apart from Agreeableness, where the monolingual performance is better. This shows that, having more data in English helps us improve the performance of our model on the cross-lingual and multilingual tasks when compared to the monolingual performance, due to limited Chinese labeled data.

Since we only use a single parallel corpus to train the bilingual word embeddings, the vocabulary size is not significantly large, when compared to that of Google word2vec for example. Therefore, our average F-score performance can be potentially improved by collecting more parallel corpora and retraining our bilingual embeddings. Also, we used a automatic speech recognition system to get the transcriptions of the Chinese dataset, for which we have to count for the ASR output error as well. Future work would involve hand correcting the transcriptions and re-training our models.

5. Future Work and Conclusion

We have collected a much larger dataset from the myPersonality project, that has around 22 Million Facebook status updates from 154,000 users labeled with personality. Such large datasets in English when trained with cross lingual word embeddings will enable us to recognize personality from texts of different languages. Also, by using a larger parallel corpus, we can increase the vocabulary size of our trained embeddings. We would like to train our bilingual word embeddings on larger and use more parallel corpora, and also include other languages to train our model on, apart from English and Chinese.

Although computational personality recognition is a widely recognized field, research on making the recognition truly multilingual and language independent is lacking in literature. Therefore, we propose a method that can make use of the relatively higher availability of a source language, such as English, with personality labels and use it to train a classifier that can identify personality from texts of other target languages, such as Chinese. Such cross-lingual and multilingual performance will in turn enable us to broaden the applications of personality recognition to multiple languages, with the limited personality labeled data that we have in certain languages across the world.

6. Acknowledgements

This work was partially funded by CERG #16214415 of the Hong Kong Research Grants Council.

7. References

[1] M. V. Long and P. Martin, “Personality, relationship closeness, and loneliness of oldest old adults and their children,” *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, vol. 55, no. 5, pp. P311–P319, 2000.

[2] D. S. Berry, J. K. Willingham, and C. A. Thayer, “Affect and personality as predictors of conflict and closeness in young adults’ friendships,” *Journal of Research in Personality*, vol. 34, no. 1, pp. 84–107, 2000.

[3] P. Fung, D. Bertero, Y. Wan, A. Dey, R. H. Y. Chan, F. B. Siddique, Y. Yang, C.-S. Wu, and R. Lin, “Towards empathetic human-robot interactions,” *arXiv preprint arXiv:1605.04072*, 2016.

[4] P. Fung, A. Dey, F. B. Siddique, R. Lin, Y. Yang, W. Yan, and R. C. H. Yin, “Zara the supergirl: An empathetic personality recognition system,” 2015.

[5] L. R. Goldberg, “The structure of phenotypic personality traits.” *American psychologist*, vol. 48, no. 1, p. 26, 1993.

[6] P. T. Costa and R. R. McCrae, “The revised neo personality inventory (neo-pi-r),” *The SAGE handbook of personality theory and assessment*, vol. 2, pp. 179–198, 2008.

[7] K.-H. Peng, L.-H. Liou, C.-S. Chang, and D.-S. Lee, “Predicting personality traits of chinese users based on facebook wall posts,” in *Wireless and Optical Communication Conference (WOCC), 2015 24th*. IEEE, 2015, pp. 9–14.

[8] J. Oberlander and S. Nowson, “Whose thumb is it anyway?: classifying author personality from weblog text,” in *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, 2006, pp. 627–634.

[9] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, “Using linguistic cues for the automatic recognition of personality in conversation and text,” *Journal of artificial intelligence research*, vol. 30, pp. 457–500, 2007.

[10] J. W. Pennebaker, M. E. Francis, and R. J. Booth, “Linguistic inquiry and word count: Liwc 2001,” *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001, 2001.

[11] M. Coltheart, “The mrc psycholinguistic database,” *The Quarterly Journal of Experimental Psychology*, vol. 33, no. 4, pp. 497–505, 1981.

[12] J. Golbeck, C. Robles, and K. Turner, “Predicting personality with social media,” in *CHI’11 extended abstracts on human factors in computing systems*. ACM, 2011, pp. 253–262.

[13] F. Celli, F. Pianesi, D. Stillwell, and M. Kosinski, “Workshop on computational personality recognition (shared task),” in *Proceedings of the Workshop on Computational Personality Recognition*, 2013.

[14] B. Verhoeven, W. Daelemans, and T. De Smedt, “Ensemble methods for personality recognition,” in *Proceedings of the Workshop on Computational Personality Recognition*, 2013, pp. 35–38.

[15] G. Farnadi, S. Zoghbi, M.-F. Moens, and M. De Cock, “Recognising personality traits using facebook status updates,” in *Proceedings of the workshop on computational personality recognition (WCPR13) at the 7th international AAAI conference on weblogs and social media (ICWSM13)*. AAAI, 2013.

[16] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” *arXiv preprint arXiv:1404.2188*, 2014.

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

[23] P. Fung and K. W. Church, “K-vec: A new approach for aligning parallel texts,” in *Proceedings of the 15th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1994, pp. 1096–1102.

[24] P. Fung and K. McKeown, “Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping.” First Conference of the Association for Machine Translation in the Americas, 1994, pp. 81–88.

[25] S. Upadhyay, M. Faruqui, C. Dyer, and D. Roth, “Cross-lingual models of word embeddings: An empirical comparison,” *arXiv preprint arXiv:1604.00425*, 2016.

[26] I. Vulic and M.-F. Moens, “Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. ACL, 2015, pp. 719–725.

[27] Y. Bengio and G. Corrado, “Bilbowa: Fast bilingual distributed representations without word alignments,” 2015.

[28] M. Faruqui and C. Dyer, “Improving vector space word representations using multilingual correlation.” Association for Computational Linguistics, 2014.

[29] T. Luong, H. Pham, and C. D. Manning, “Bilingual word representations with monolingual quality in mind,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 151–159.

[30] L. Tian, D. F. Wong, L. S. Chao, P. Quaresma, F. Oliveira, and L. Yi, “Um-corpus: A large english-chinese parallel corpus for statistical machine translation.” in *LREC*, 2014, pp. 1837–1842.

[31] M. Kosinski, S. C. Matz, S. D. Gosling, V. Popov, and D. Stillwell, “Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines.” *American Psychologist*, vol. 70, no. 6, p. 543, 2015.

[32] J.-I. Biel and D. Gatica-Perez, “The youtube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs,” *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 41–55, 2013.

[33] M. Kosinski and D. Stillwell, “mypersonality research wiki,” 2012.

[34] Y. Zhang, J. Liu, J. Hu, X. Xie, and S. Huang, “Social personality evaluation based on prosodic and acoustic features.” International Conference on Machine Learning and Soft Computing, 2017.

[35] Y. Goldberg and O. Levy, “word2vec explained: Deriving mikolov et al.’s negative-sampling word-embedding method,” *arXiv preprint arXiv:1402.3722*, 2014.

[36] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.