Learning Digital Geographies through a stacked Multi-Modal Autoencoder

Pengyuan Liu[*1], Stefano De Sabbata[†1]

[1]University of Leicester

September 7, 2018

Summary

As social media have become an integral part of most people's everyday life, there has been an increasing interest in exploring and identifying users' opinions, trends, and popular events in both physical and online world within industry and academic research. In this paper, we introduce an unsupervised clustering approach to cluster social media posts from Twitter based on their text, image, and geo-location. Our approach combines a stacked multi-modal autoencoder neural network to create joint representations of text and image, and canonical-correlation analysis (CCA) and a hierarchical clustering algorithm.

KEYWORDS: Twitter, multimodal autoencoder, neural network, CCA

1. Introduction

Twitter provides a vast repository of human perspectives and sentiments regarding a broad spectrum of topics by millions of users. Recent years have seen a growing interest in the analysis of such information, including the development of the field of digital geography (Ash et al., 2018). However, due to the vast amount of data produced daily on social media, qualitative analysis can only tackle samples of such data, and quantitative analysis and summarisation are frequently a necessary step in digital geography.

This creates a strong point of connection with GIScience, where clustering is a key approach to identify users' opinions and trends, to study the emergence of place from space through content creation, or to monitor events from football to earthquakes (see e.g., Frias-Martinez and Frias-Martinez, 2014; Ifrim et al., 2014; Sechelea et al., 2016; Zahra et al., 2017). In computer science, sentence-level topic extraction from social media has also been an important research topic in the last ten years, and research as mainly focused on supervised learning approaches with labelled data (Medhat et al., 2014). However, labelling large volumes of Twitter datasets requires human intervention, can become costly, and it is viable only if a pre-defined set of topics or categories is identified.

Less attention has been given so far to exploratory analysis scenarios, where only vague categories or no categories at all have been pre-defined. That is, how can we identify unknown events and trends in social media, beyond simple hashtags-based analysis? Unsupervised machine learning techniques, which don't require labelled data, and approaches such as clustering can perform topic extraction by grouping tweets based on their semantic similarities. In this paper, we introduce an unsupervised approach based on a stacked multi-modal autoencoder and a hierarchical clustering algorithm, clustering tweets based on their textual content, image, and geo-location.

[*] pl164@leicester.ac.uk

[†] s.desabbata@leicester.ac.uk

2. Methodology

Inspired by the Deep Embedding Clustering (DEC) approach introduced by Xie et al. (2016), we propose an approach consisting of three components, illustrated by the flow chart in **Figure 1**. First, a stacked autoencoder model (e.g., Guo et al., 2017) is used to extract dense representations from both texts and images of tweets. Second, a canonical-correlation analysis (CCA) is applied to the extracted dense representations and geo-coordinates to find the maximum correlation between the contents of the tweets and their geo-locations. Finally, an agglomerative hierarchical clustering algorithm is employed to generate the final cluster.

2.1. Multi-Modal Autoencoder

Our proposed multi-modal autoencoder (shown in **Figure 2)** consists of five major parts: two encoders for images and texts respectively, and their two corresponding decoders, plus a layer to calculate maximum correlation coefficient stacked after the concatenation of image and text representations. This approach is inspired by the Correlational Neural Network (Cornet) introduced by Chandar et al. (2013), which learns joint representations by maximising correlation of two views when projected to common subspace. The dense layers are replaced with a Resnet-style convolution layers for image representations (Ledig et al., 2017) and an LSTM layers for textual representations. The latter adopts a text summarisation architecture proposed by Lopyrev (2015) for the encoder and reconstructs nouns and adjectives of each text in its decoder. The objective is not only to minimise the self-construction error, but also the cross-reconstruction error from image and texts, and maximise the correlation between the hidden representations of both parts. We achieve this by minimising the objective function introduced in the original Cornet paper. We use squared error loss as the reconstruction error.

2.2. Canonical-Correlation Analysis and Hierarchy Clustering

Having created a dense, numeric representation from each tweet's image and text from the encoder parts of our multi-modal autoencoder, we perform Kernel CCA on those representations and the corresponding geo-locations. We first employed principal component analysis (PCA) to reduce the dimension of the extracted representations from 399424 to 98 with 98.2% information preserved, and then t-Distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimension from 98 to 2. The Kernel CCA algorithm is thus used to create new 2-dimensional representations. Finally, an agglomerative hierarchical clustering algorithm is used to cluster the newly created representations.

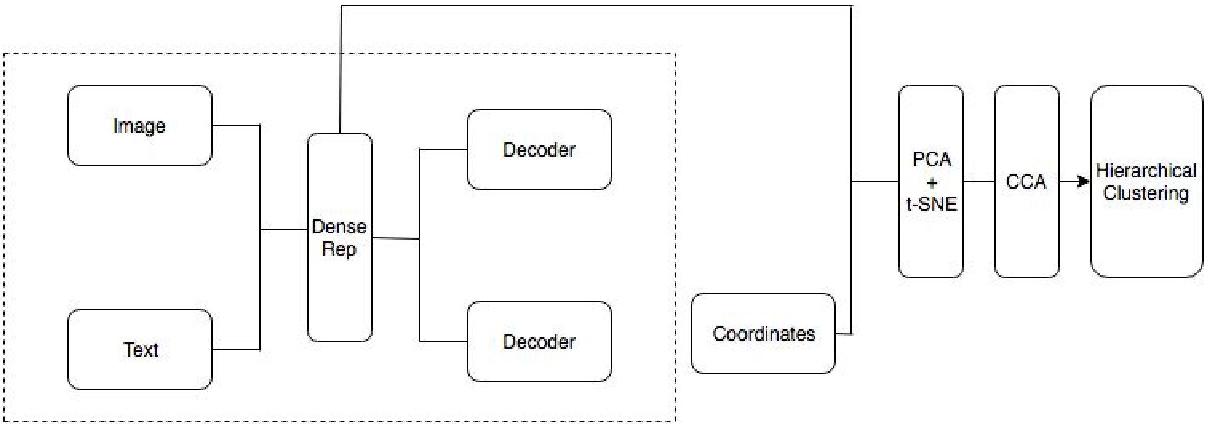

Figure 1 - Methodology Flowchart

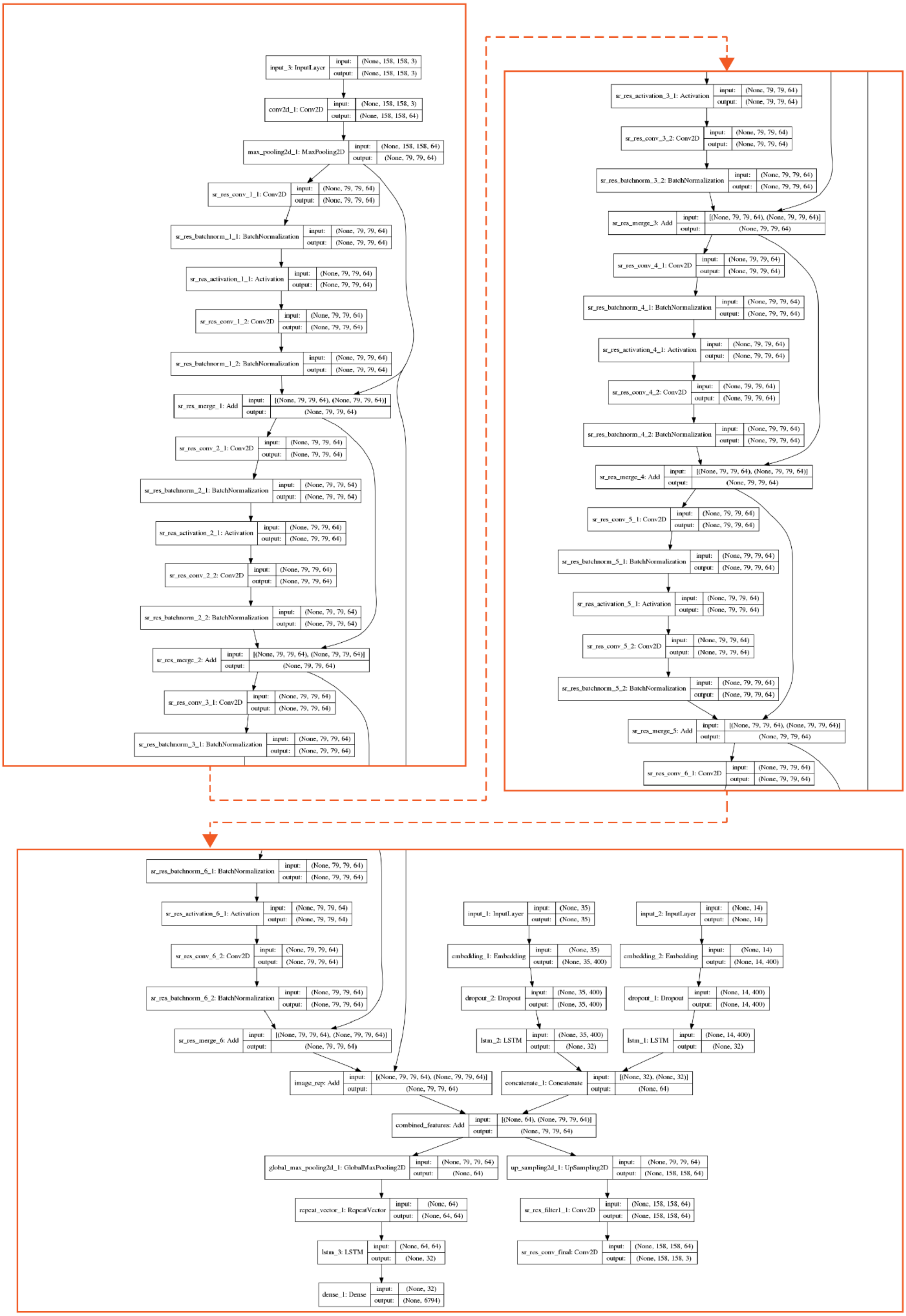
input_3: InputLayer input: (None, 158, 158, 3) output: (None, 158, 158, 3)
conv2d_1: Conv2D input: (None, 158, 158, 3) output: (None, 158, 158, 64)
max_pooling2d_1: MaxPooling2D input: (None, 158, 158, 64) output: (None, 79, 79, 64)
sr_res_conv_1_1: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_1_1: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_activation_1_1: Activation input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_conv_1_2: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_1_2: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_merge_1: Add input: [(None, 79, 79, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
sr_res_conv_2_1: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_2_1: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_activation_2_1: Activation input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_conv_2_2: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_2_2: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_merge_2: Add input: [(None, 79, 79, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
sr_res_conv_3_1: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_3_1: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_activation_3_1: Activation input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_conv_3_2: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_3_2: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_merge_3: Add input: [(None, 79, 79, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
sr_res_conv_4_1: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_4_1: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_activation_4_1: Activation input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_conv_4_2: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_4_2: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_merge_4: Add input: [(None, 79, 79, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
sr_res_conv_5_1: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_5_1: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_activation_5_1: Activation input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_conv_5_2: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_5_2: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_merge_5: Add input: [(None, 79, 79, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
sr_res_conv_6_1: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_6_1: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_activation_6_1: Activation input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_conv_6_2: Conv2D input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_batchnorm_6_2: BatchNormalization input: (None, 79, 79, 64) output: (None, 79, 79, 64)
sr_res_merge_6: Add input: [(None, 79, 79, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
input_1: InputLayer input: (None, 35) output: (None, 35)
input_2: InputLayer input: (None, 14) output: (None, 14)
embedding_1: Embedding input: (None, 35) output: (None, 35, 400)
embedding_2: Embedding input: (None, 14) output: (None, 14, 400)
dropout_2: Dropout input: (None, 35, 400) output: (None, 35, 400)
dropout_1: Dropout input: (None, 14, 400) output: (None, 14, 400)
lstm_2: LSTM input: (None, 35, 400) output: (None, 32)
lstm_1: LSTM input: (None, 14, 400) output: (None, 32)
image_rep: Add input: [(None, 79, 79, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
concatenate_1: Concatenate input: [(None, 32), (None, 32)] output: (None, 64)
combined_features: Add input: [(None, 64), (None, 79, 79, 64)] output: (None, 79, 79, 64)
global_max_pooling2d_1: GlobalMaxPooling2D input: (None, 79, 79, 64) output: (None, 64)
up_sampling2d_1: UpSampling2D input: (None, 79, 79, 64) output: (None, 158, 158, 64)
repeat_vector_1: RepeatVector input: (None, 64) output: (None, 64, 64)
sr_res_filter1_1: Conv2D input: (None, 158, 158, 64) output: (None, 158, 158, 64)
lstm_3: LSTM input: (None, 64, 64) output: (None, 32)
sr_res_conv_final: Conv2D input: (None, 158, 158, 64) output: (None, 158, 158, 3)
dense_1: Dense input: (None, 32) output: (None, 6794)

Figure 2 - The proposed multi-modal autoencoder.

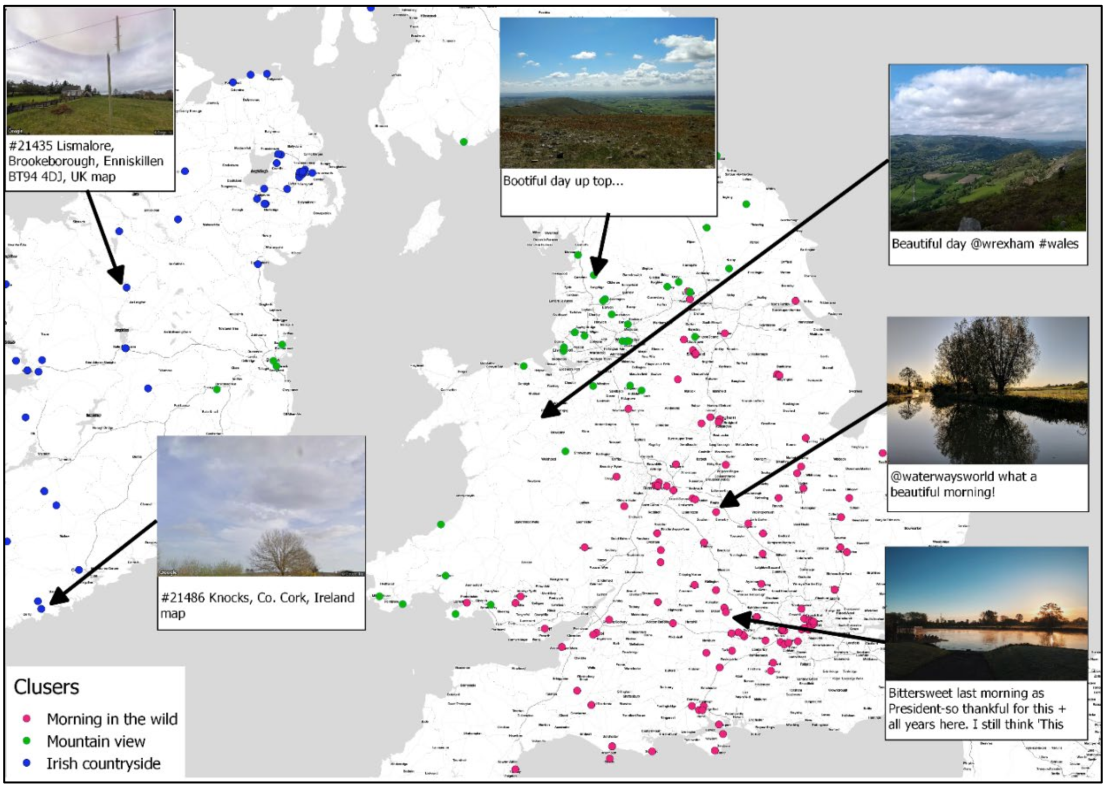
#21435 Lismalore, Brookeborough, Enniskillen BT94 4DJ, UK map
Bootiful day up top...
Beautiful day @wrexham #wales
@waterwaysworld what a beautiful morning!
#21486 Knocks, Co. Cork, Ireland map
Bittersweet last morning as President-so thankful for this + all years here. I still think 'This
Clusers
● Morning in the wild
● Mountain view
● Irish countryside

Figure 3 – Three of the 42 identified clusters and example tweets.

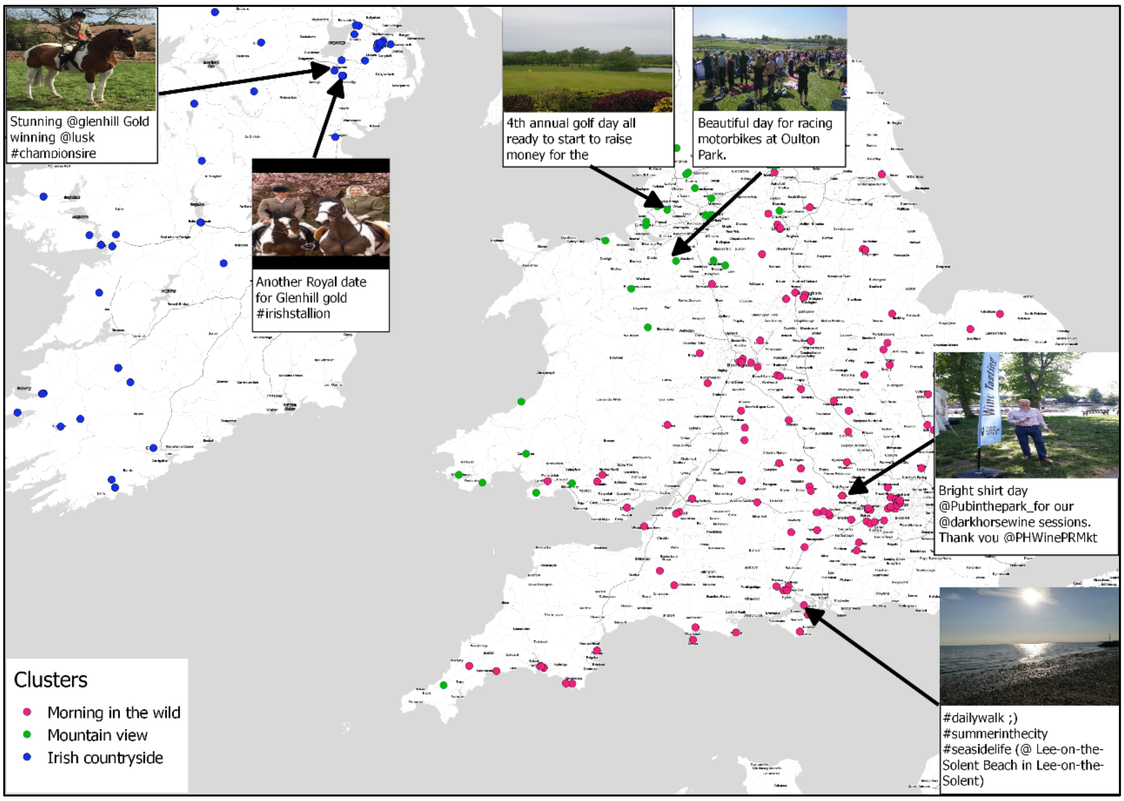
Stunning @glenhill Gold winning @lusk #championsire
4th annual golf day all ready to start to raise money for the
Beautiful day for racing motorbikes at Oulton Park.
Another Royal date for Glenhill gold #irishstallion
Bright shirt day @Pubinthepark_for our @darkhorsewine sessions. Thank you @PHWinePRMkt
#dailywalk ;) #summerinthecity #seasidelife (@ Lee-on-the-Solent Beach in Lee-on-the-Solent)
Clusters
● Morning in the wild
● Mountain view
● Irish countryside

Figure 4 – Examples of cluster not entirely conforming to the rest of the cluster.

3. Experiment

We evaluate our model on a dataset extracted of social media posts collected through the Twitter API[‡] between 7th May 2018 and 20th May 2018 in the British Isles. We selected tweets containing images, texts, and exact coordinates, and limiting the number of tweets per account to 10, obtaining a dataset of 2876 tweets.

3.1. Word Embeddings and Hyper-parameters

We performed tokenisation, stopwords removal and case folding on the texts. We used an existing collection of word embeddings pre-trained with Twitter data[§] to generate a 400-dimension vector representation of each text. The proposed multi-modal autoencoder model uses 32 hidden units for all LSTM layers, and 64 hidden units for all convolutional layers, using a Rmsprop optimiser with 0.001 learning rate. Each image was converted into a grey scale and resized to158×158 pixels.

4. Results and Future Work

We applied the approach introduced in Section 2 to the data described in Section 3, obtaining 42 clusters. **Figure 3** illustrates three of the 42 clusters**,** which could be understood as focusing on: morning in the wild, mountain view and Irish countryside.

A preliminary analysis of the results seems to indicate that the proposed methodology has the ability to capture both content similarities of images and texts and geographical closeness. However, each cluster contains some level of noise, as illustrated in **Figure 4**. The use of slangs, misspellings, little-used hashtags constitute a major source of noise in the clustering and establishing a semantic link between texts and images is a significant challenge for our model. These results seem to indicate that further data pre-processing might be necessary to improve the model.

The project is still in its early stage of development, and formal assessment of the results is beyond the scope of this short abstract. We are currently planning to test our approach by comparing our results to clusters created by human participants to an experiment that is still under development, possibly through a crowdsourcing platform such as Mechanical Turk. Further developments might include substituting the CCA component with a graph net (Battaglia et al., 2018).

References

Ash, J., Kitchin, R. and Leszczynski, A., 2018. Digital turn, digital geographies?. *Progress in Human Geography*, *42*(1), pp.25-43.

Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R. and Gulcehre, C., 2018. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.

Chandar, S., Khapra, M.M., Larochelle, H. and Ravindran, B., 2016. Correlational neural networks. *Neural computation*, *28*(2), pp.257-285.

Frias-Martinez, V. and Frias-Martinez, E., 2014. Spectral clustering for sensing urban land use using Twitter activity. Engineering Applications of Artificial Intelligence, 35, pp.237-245.

Guo, X., Liu, X., Zhu, E. and Yin, J., 2017, November. Deep clustering with convolutional autoencoders. In *International Conference on Neural Information Processing* (pp. 373-382). Springer, Cham.

[‡] https://developer.twitter.com/en/docs.html

[§] https://github.com/loretoparisi/word2vec-twitter

Ifrim, G., Shi, B. and Brigadir, I., 2014, April. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. ACM.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z. and Shi, W., 2017, July. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *CVPR* (Vol. 2, No. 3, p. 4).

Lopyrev, K., 2015. Generating news headlines with recurrent neural networks. *arXiv preprint arXiv:1512.01712*.

Medhat, W., Hassan, A. and Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), pp.1093-1113.

Sechelea, A., Do Huu, T., Zimos, E. and Deligiannis, N., 2016, May. Twitter data clustering and visualization. In ICT (pp. 1-5).

Xie, J., Girshick, R. and Farhadi, A., 2016, June. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (pp. 478-487).

Zahra, K., Ostermann, F.O. and Purves, R.S., 2017. Geographic variability of Twitter usage characteristics during disaster events. *Geo-spatial information science*, *20*(3), pp.231-240.

Biographies

Pengyuan Liu is a third year PhD student at the University of Leicester, with research interests regarding Artificial Intelligence in GIScience, practically in developing deep learning methodologies to better understand digital geographies.

Stefano De Sabbata is Lecturer in Quantitative Geography at the School of Geography, Geology and the Environment of the University of Leicester and Research associate of the Oxford Internet Institute of the University of Oxford.