Utilizing External Knowledge to Enhance Location Prediction for Twitter/X Users in Low Resource Settings

YAGUANG LIU, Georgetown University, Washington, United States
LISA SINGH, Georgetown University, Washington, United States

Accurate estimates of user location are important for many online services, including event detection, disaster management, and determining public opinion. Neural network-based techniques have proven to be highly effective in predicting user location. However, these models typically require a large amount of labeled training data, which can be difficult to obtain in real-world scenarios. In this article, we present two approaches to tackle the issue of limited training data when predicting city level location. First, we consider a self-supervised approach that trains a state-level model without labeled data and then integrate this knowledge into the training dataset used for city-level predictions. Second, we explore the option of increasing the number of training examples by utilizing external resources to generate *synthetic users*. Finally, we combine these two strategies, exploiting the benefits of both. We empirically evaluate our proposed techniques on multiple Twitter/X datasets and show that our models perform significantly better than the state-of-the-art with improvements of up to 6% for Acc@161 and 8% for F1 score.

CCS Concepts: • **Computing methodologies → Machine learning;**

Additional Key Words and Phrases: Location inference, limited training data, self supervised learning

ACM Reference Format:
Yaguang Liu and Lisa Singh. 2024. Utilizing External Knowledge to Enhance Location Prediction for Twitter/X Users in Low Resource Settings. *ACM Trans. Spatial Algorithms Syst.* 10, 3, Article 19 (July 2024), 25 pages. https://doi.org/10.1145/3673899

1 Introduction

Social media platforms like Twitter/X have emerged as a significant source of real-time breaking news, event sharing, and public opinion. Associating users' location to shared content is useful for many applications, including disaster response, public health monitoring, urban planning, and marketing [30, 42, 53, 55]. For example, when a natural disaster occurs, emergency response teams can more rapidly help communities if they know the location of users who post about on-the-ground conditions [9, 53, 54]. Social scientists can also utilize posts with location information to understand public opinion about issues related to elections, the economy, and well-being. In

This work was supported by National Science Foundation awards #1934925 and #1934494, the National Collaborative on Gun Violence Research (NCGVR), and the Massive Data Institute (MDI) at Georgetown University.
Authors' Contact Information: Yaguang Liu, Georgetown University, Washington, District of Columbia, United States; e-mail: yl947@georgetown.edu; Lisa Singh, Georgetown University, Washington, District of Columbia, United States; e-mail: Lisa.Singh@georgetown.edu.

ACM 2374-0353/2024/07-ART19
https://doi.org/10.1145/3673899

these cases, knowing the location of Twitter/X users is necessary for constructing a representative sample [6, 55].

Despite the need, obtaining reliable location information about different subpopulations of interest remains challenging for two primary reasons. First, even though Twitter/X allows users to share their locations either in the location field or through GPS tracking, a large fraction of users choose to do neither. For example, it has been reported that less than 1% of Twitter/X posts have geolocation information associated with them [31]. Second, even when users share location information, it is often incomplete and inaccurate [27]. Manual labeling has also proven to be difficult, because annotators often cannot reach a consensus, resulting in limited training data, especially when the analysis includes a large number of cities [11, 49]. Consequently, researchers often encounter scenarios where they have to rely on model predictions trained using small datasets. It is not atypical for studies involving manual annotation for training data to have a small amount of labeled data, ranging from hundreds to thousands of users.

Because of the importance of location information, much research on location inference exists [23, 24, 30, 41, 42]. These methods typically work by training a model using a hybrid set of features, including tweet text, profile, and network information. While these methods are powerful when large amounts of training data are available, they are insufficient when the training data are limited. Therefore, work still remains in this low-resource setting, i.e., when the amount of labeled training data is limited and each location in the training set has location labels for only a small number of users (dozens or even less). Prior work has demonstrated that the performance of deep learning models for different text classification tasks degrades significantly in this case [20, 37].

To address these challenges, this article focuses on city-level location inference in a low-resource setting. We use tweet text and, when available, the location field text (self-reported location).[1] We posit that to address this challenge, there are three broad directions we can take: (1) provide better user representations through pretraining instead of training from scratch; (2) expand the number of training examples by generating larger quantities of high-quality labeled training data; (3) utilize a combination of both strategies. In our work, we test these different directions. To improve user representations, we incorporate **self-supervised learning (SSL)** into our pretraining. SSL is a machine learning process where the model trains itself to learn meaningful representations of input data by creating auxiliary tasks and generating data labels for that auxiliary task automatically. For this city location detection task, we propose *Related Concept Pretraining* as a form of SSL. The general idea is to pretrain a model for a similar task on a random set of users and then add this additional knowledge to the users in the training set by fine-tuning their vector representations. To accomplish this, we construct a dataset that can be used for pretraining our model. This dataset uses self-supervised learning to generate a state location without human annotation. We refer to this new self-supervised signal as a *pseudo-label*. A model is pretrained using the state location pseudo-label and then incorporated as additional knowledge into the training process of the city prediction.

To increase the number of training examples, we propose *Data Augmentation Using External Resources*. Specifically, we take advantage of external knowledge that exists online to automatically generate labeled data and then inject these examples into our training data. Some research [3, 34] has demonstrated that leveraging location-related text found online can help with location inference. Our data-augmentation approach explores this idea. Existing models typically rely on identifying indicative words as guidance for model training. However, when applying deep learning techniques to Twitter/X data, these approaches tend to under-perform because of the

[1]We do not adopt the commonly used mention network both for privacy reasons (it involves processing users' handles) and the resource constraint focus of our problem. Other types of data, such as timezone, are currently unavailable through the Twitter/X API we have access to.

Fig. 1. A tweet discussing crime in Houston from a user who lives there. It does not include the city name, but the details shared align closely with what is found on Wikipedia.

inherent noisiness of tweets and the limited effectiveness of word-level analysis [4, 38]. We take a different approach. We take sentence and paragraph-level context from location-relevant pages and generate *synthetic users* to add into our training dataset. We hypothesize that contextually relevant words, sentences, or paragraphs can be informative. For example, Figure 1 shows a hypothetical tweet mentioning information about Houston's crime rate. The city name is not present in the tweet. However, the Wikipedia page for the city of Houston states "Houston's violent crime rate was 8.6% percent higher in 2016 than the previous year; however, from 2006 to 2016, violent crime was still down 12 percent in Houston" [59]. If the Wikipedia knowledge is part of the model, then the content of the tweet without the explicit city name may imply a user's location. Given this observation, our approach involves creating synthetic users whose posts are sourced from location-related articles and assigning the corresponding locations to the users as labels. This method effectively enlarges the size of the training data with valuable information and, as we will show, it generally improves performance. Finally, we investigate the performance of combining Related Concept Pretraining and Data Augmentation using External Resources, with the objective of leveraging the strengths of both approaches to improve the accuracy and robustness of the model.

While our primary interest is in developing location inference models given a resource constrained environment, we also have concerns about the lack of high-quality labeled data, i.e., the reliability of ground truth datasets for the location inference task. For example, despite the popularity of using geotagged information as the ground truth, different public datasets have different rules for determining the users' labels. Example strategies include using the coordinates of the first geotagged tweet as the location of a user or using a majority vote of the collected tweets. Given the variability in the way ground truth datasets are constructed, we explore the robustness and validity of these methods by comparing different geotag labeling strategies to self-reported locations from users' profiles. Ultimately, we want to understand the impact of different ground-truthing strategies and provide recommendations that may improve the consistency of research in this area.

In summary, we present a neural network-based model for city-level user location detection on Twitter/X in a low-resource setting and make the following contributions: (1) We propose self-supervised and data-augmentation approaches for advancing location inference on Twitter/X. (2) We use Related Concept Pretraining to integrate state-level knowledge into the training process as a strategy for improving the quality of the training data for this specific task. (3) As a complementary strategy, we propose using location-related articles, e.g., Wikipedia pages, as external knowledge to increase the number of examples in the training data without manual labeling. (4) We conduct an extensive empirical analysis and demonstrate the strengths of each strategy on different Twitter/X datasets. (5) We present an analysis of the integrated approach that combines both external knowledge pretraining and training data expansion and show its effectiveness. (6) We investigate the most commonly used ground truth labeling strategies and demonstrate the strengths and weaknesses of each. (7) We make our code publicly available to support research in the area.[2]

[2]The code can be found through https://github.com/GU-DataLab/location_prediction

The remainder of this article is organized as follows: In Section 2, we review the relevant literature. In Section 3, we present our experimental design. Section 4 describes our datasets. In Section 5, we present our empirical evaluation. Section 6 presents conclusions and future work.

2 Related Literature

The related literature is divided into two parts. First, we describe previous work on location inference (Section 2.1) and demographic inference (Section 2.2. Then, we review approaches for addressing challenges that arise when using limited training data (Section 2.3).

2.1 Location Inference

Location information, such as countries, states, or cities, plays a vital role in contextualizing news, emergency events, and people's behaviors. Given its importance, automatic identification of locations has been studied for decades [63].

Location Categories. There are many different labels used for location inference tasks. In most studies, home locations are predicted at city-level [30, 41, 42], and sometimes the classification task is at the state, country, points, or grids level [27, 50, 51]. State-level categorization typically uses all the states in a specific country as label categories, and country-level categorization involves using a specific country as the category. For points-level, latitude and longitude are used as the location categories. For grid categorization, the surface of the earth is represented as a two-dimensional space over latitude and longitude pairs. Categories are created by grouping pairs whose latitude and longitude are close to each other. City-level classification uses city labels to represent each city and the immediate surrounding suburb. This strategy is used because of the large population variation between large and small cities [23]. To mitigate this issue, most studies follow the city category method of Han and colleagues [23], grouping together large cities with their surrounding satellites and suburb cities and using them as labels.

Location Inference Approaches. The earliest attempts to identify the locations of users involved mapping their IP addresses to physical locations [8], but this approach relies on private information only accessible to internet service providers. Subsequent efforts focused on text-based methods. Amitay and colleagues [3] propose a model that extracts information related to a location listed in a gazetteer to identify geographical regions of web pages. Bilhaut et al. [5] build a rule-based geographical classifier that utilizes a geographical gazetteer as an external lexicon.

Researchers have explored the use of classic machine learning algorithms like **Support Vector Machine (SVM)** for location inference of users on social media platforms [12, 39]. Han and colleagues [23] propose using location-indicative words via feature selection, e.g., maximum entropy, for this task. Han and colleagues [24] also propose a stacking-based method that combines tweet text and metadata, including self-reported location and time zone. Krishnamurthy and colleagues [34] introduce a knowledge-based approach using Wikipedia to improve location classification accuracy. In their model, they match the entities of a city from Wikipedia and the entities mentioned by the user in his/her posts and then predict the most likely location of a user.

With the increasing popularity of deep learning methods, numerous neural network-based methods have been proposed. Miura and colleagues [41] map text, self-reported location, biographies, and timezone into the embedding space using fastText [7] and then concatenate all the embeddings from different components. This model achieves a comparable performance to some of the advanced models that employ classic machine learning methods. Huang and Carley [29] propose a CNN-based model that uses both text and metadata. Networks have also been investigated in the context of deep learning for location prediction by Ebrahimi and colleagues [17]. In their work, the authors construct a graph from users who mention each other and use network embeddings

to represent users. The main limitation of network-based models is that users who are not linked to the network display suboptimal performance in the test set. Miura and colleagues [42] propose a GRU model that combines tweet text, biography, and network embeddings using an attention mechanism, and their model outperforms several baselines. A semi-supervised learning approach is proposed by Rahimi and colleagues [48], but their model does not perform as well as the state-of-the-art in many cases. Mahmud and colleagues [39] develop a two-level hierarchical location classifier that predicts a country location mapped from the city label and then the city label within the former. Wing and Baldridge [60] also build a hierarchical tree as the classification structure. Both of these methods require training one classifier separately many times, reducing their efficiency.

Huang and Carley [30] introduce a hierarchical location prediction neural network, which simultaneously predicts a coarse-grained location (state) and the fine-grained label (city), the former of which serves as a guide of the latter and thus, greatly reduces the training time. They employ both word-level and character-level embedding in their model and create a new state-of-the-art. Despite this improvement, we will show that when dealing with small datasets, the model still faces challenges in a low-resource setting when there are a limited number of labeled examples available for training at both the city-level and state-level. Our work differs from these previous works, since we pretrain a model using generated state-level pseudo-labels that were obtained from a large number of arbitrary users on Twitter/X. These pseudo-labels are also added as additional features to our training examples. We also use external knowledge to generate new user training examples to increase the size of our training data. To the best of our knowledge, SSL has not been proposed for location detection to enhance user representations, and using available documents to create synthetic users to increase the sample size has also not been explored.

2.2 Other Demographics

Inferring individuals' attributes such as gender and age on Twitter/X has been used to understand public opinion of different groups and conduct social science research on a variety of topics such as harassment and health [16, 28]. As a result, the inference of a Twitter/X user's demographics has become an active area of research.

Earlier work often focused on using classic algorithms such as **Naive Bayes (NB)** with features generated from bag-of-words [11, 44, 52]. Rao and colleagues [49] introduce sociolinguistic features, i.e., the use of words such as LOL and OMG, and they achieve higher accuracy than using n-grams alone when inferring gender, age, and political orientation. Fink and colleagues [21] utilize n-grams, hashtags, and LIWC features to build an SVM classifier for gender inference, but the performance improvement is marginal. Chen and colleagues [11] employed a more extensive set of features that combines n-grams, topic features, profile images, and names to achieve state-of-the-art performance [11] for age and gender inference. Zamal and colleagues [2] propose using features generated from networks for political orientation. Specifically, they adopt features such as n-grams and statistics and also the same feature set from users' friends. Their SVM model performs significantly better than the bag-of-words-based model.

In recent years, deep neural network-based models have attracted much attention. Kim and colleagues present a model using word embeddings and **recursive neural networks (RNN)** for gender and age inference. Wood-Doughty and colleagues propose using names only (last name and screen name) for gender inference [61]. In their work, the models encode names and screen names using **convolutional neural network (CNN)** and RNN, separately. Their best model achieves a higher result than classic machine learning models. A multi-modal model is proposed by Vijayaraghavan and colleagues [57], where they use an attention mechanism [14] to combine components, including profile image, tweet text, and user network. Their model significantly

enhances the accuracy for inference of age, gender, and political orientation when compared to the traditional models. The rise of large-scale pretrained models has led to the growing popularity of sentence-level embeddings. Liu and colleagues [38] propose using BERT for gender and age classification, and the results show that sentence-level embeddings perform much better than word-level analysis. A hierarchical network-based model developed by Liu and Singh [36] uses tweet text and emojis for age inference. Specifically, they use **gated recurrent unit (GRU)** [15] with an attention layer to separately train the emoji component (using word embedding and CNN) and the text component (using BERT). **Contrastive Language-Image Pre-training (CLIP)** [47] is introduced by Liu and Singh for age and gender prediction, and the accuracy is shown to improve greatly over BERT [36]. The limitation of all these neural models is that they require large amounts of labeled data (typically thousands per label) for pretraining and/or fine-tuning.

2.3 Algorithms for Small Training Data

Research in this space focuses on three strategies for handling small training datasets: transfer learning, data augmentation, and self-supervised learning.

2.3.1 Transfer Learning. Transfer learning is a machine learning approach with the goal of enhancing performance on a related task by utilizing the knowledge gained from a previously learned task. It has become an important approach for enhancing deep learning models on small datasets. For example, Mou and colleagues [43] use transfer learning with a convolutional neural network for sentence entailment recognition. In their work, they pretrain a model using a large labeled dataset and fine-tune it for two smaller datasets to achieve better performance. Agrawal and Awekar [1] apply transfer learning for cyberbullying detection, and they experiment with transferring embedding weights and network weights from one model to another. This approach results in significant improvement in accuracy. Transfer learning has also been proposed for demographic inference. For example, Liu and Singh [37] adopt supervised transfer learning for gender and age inference on Twitter/X by pretraining a model with a large labeled dataset and achieve significant improvement over the state-of-the-art. One of the main drawbacks of these types of transfer learning methods is that they typically assume a (large) labeled dataset for pretraining, which is not always feasible, since manual labeling is time-consuming and resource-intensive.

2.3.2 Data Augmentation. Traditionally, data augmentation is a technique that involves generating new data points from existing data to increase the size of a dataset. The goal of data augmentation is to introduce additional variance and diversity into the dataset, which can improve the robustness of learning models. For image augmentation, common techniques include flipping, rotating, cropping, and translating images [40, 46]. For text, popular methods include altering text data through methods such as word replacement or shuffling. For example, Wei and Zou [58] propose **easy data augmentation (EDA)** and show that by augmenting data with techniques such as synonym replacement and random word insertion, the performance of numerous NLP tasks improved. Karimi et al. [32] propose ***An Easier Data Augmentation*** **(AEDA)**, which inserts punctuation marks into the input sequence. They improve the state-of-the-art on different news classification tasks. Back-translation, which involves translating a sentence from one language to another and then translating it back to the original language, has also shown some success [26]. Our approach for data augmentation is novel, since we use external knowledge to increase the number of examples in the training set. One limitation of data augmentation is that it can potentially result in a large distributional discrepancy between the original text and the augmented text, leading to a negative impact on the performance of the model. Therefore, when employing data augmentation, distributional alignment must exist between the original and augmented texts to see a performance gain.

2.3.3 Self-Supervised Learning (SSL). Self-supervised learning is another popular method for addressing issues caused by small data. It has recently drawn attention because of its generalization ability and has been used across different domains, including natural language processing [19, 35] and computer vision [10, 22]. For instance, **Bidirectional Encoder Representations from Transformers (BERT)**, which has been shown to be effective on a wide range of NLP tasks, is pre-trained on two tasks: masked language modeling, where a model is trained to predict as random sample of input tokens that have been replaced by a placeholder, and next sentence prediction, where the goal is to decide whether a pair of sentences are related and whether one sentence is the subsequent sentence of the other. RoBERTa [35] was introduced as a refinement of the BERT model. The authors show that BERT's performance can be greatly enhanced with improved training strategies and data-processing techniques such as training models longer, larger batches, and more training data. However, these models are often at the post-level, limiting the number of tokens in the input. Our task is a user-level task, so we can take advantage of using multiple posts (tweets) to represent one training example. Feng and colleagues [19] develop a self-supervised approach for bot detection using the number of followers as the prior knowledge for pretraining. Their model combines tweets, user profile information, and network embeddings using an attention mechanism that utilizes word embedding to encode text and profile information. This strategy has been shown to not be as effective at dealing with the noisiness of tweets [37]. While these strategies have performed well, these problems typically have a small number of classes that are being predicted—less than 10. Our task, city-level prediction. contains hundreds of classes. In general, our approach attempts to use SSL to enhance the vector representations of users in the training data and data augmentation to generate new training examples. In both cases, our approaches are novel for demographic inference and location inference.

3 Methods

In this section, we present our proposed approaches for location inference in a resource-constrained environment. We begin with a problem formulation (Section 3.1). Then, we describe our base model for location prediction (Section 3.2). Finally, we introduce the proposed methods (Sections 3.3 and 3.4).

3.1 Problem Formulation

Suppose we have a set of users U and a subset of the users are part of a labeled training dataset U^T, where $U^T = \{u_1^T, u_2^T, \ldots, u_m^T\}$ and u_i^T represents the ith user in the training dataset and m representing the number of users in the training dataset. Each user u_i^T is represented by a set of features x_i^T. The features may be tweet text or self-reported location. We denote this as $x_{i,text}^T$ and $x_{i,loc}^T$, respectively. We are interested in predicting the city label y_i^T for each u_i. Therefore, at a broad level, our goal is to build a neural model N to predict location $Y = \{y_1^T, y_2^T, \ldots, y_n^T\}$ using features $X^T = \{x_1^T, x_2^T, \ldots, x_n^T\}$ of our users U^T: $N(X^T) = Y^T$. In cases where there is no ambiguity, we will simplify the notation by removing the training set T superscript: $N(X) = Y$.

3.2 Base Model

Our approach is to enhance both the learning model and the training data. We begin by describing the base learning model we use. Figure 2 shows the overview of this model. This base model first maps tweet text and self-reported location into separate embedding spaces. Then, each type of data is input into an RNN with attention. N_{text}^T is used to process tweet text embeddings in the *Tweet Text Component*, and N_{loc}^T is used to process the location field representations in the *Location Field Component*. Finally, an attention layer combines the information to make the city prediction.

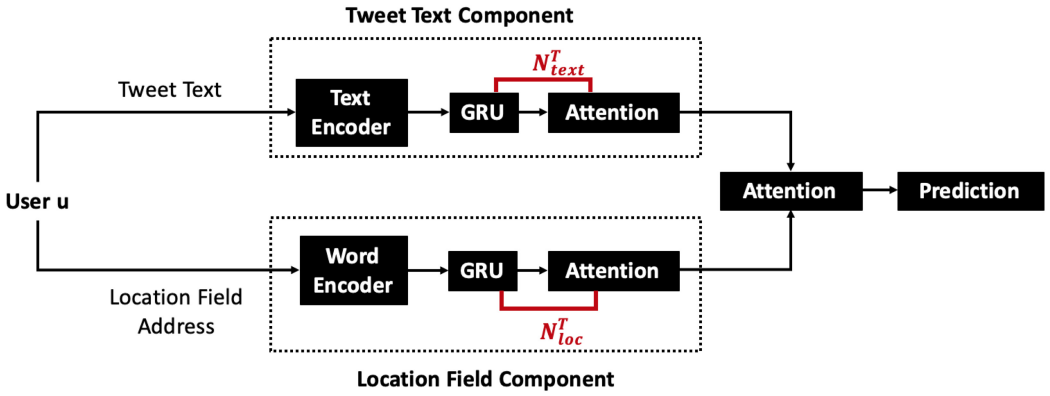


Fig. 2. Overview of the base model N.

3.2.1 Tweet Text Component. To get user post representations, we first use a sentence encoder to encode the tweet text X^T_{text} into an embedding space.

$$s_i = sentence\_encoder\left(x^T_{i,text}\right), i \in [1, m]$$

Then, we adopt a **Hierarchical Attention Network (HAN)** [62] as the structure of N^T_{text}, which includes a GRU with an attention network. Compared to traditional attention network, HAN is able to capture the intra-post relationship between these different post components, as well as the inter-post relationships of a user's posts. This structure has been shown to be useful for user-level tasks with Twitter/X data [37]. Specifically, we use a bidirectional GRU to encode tweet text representations:

$$\overrightarrow{h_i} = \overrightarrow{GRU}(s_i), i \in [1, m]$$
$$\overleftarrow{h_i} = \overleftarrow{GRU}(s_i), i \in [m, 1].$$

We concatenate $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ to get an annotation of the tweet text for user i, $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$. Next, we use an attention mechanism to reward tweet text that provides more important features for correctly classifying a user. This yields

$$f_i = tanh(Wh_i + b),$$
$$\alpha_i = \frac{exp(f_i^T o)}{\sum_i exp(f_i^T o)},$$
$$v = \sum_i \alpha_i h_i.$$

The tweet annotation h_i is fed through a **MultiLayer Perceptron (MLP)** to get f_i as a hidden representation. Then, we measure the importance of the tweet text with a context vector o and get a weight α_i through a softmax function. Finally, we compute the tweets vector v as a weighted sum.

3.2.2 Location Field Component. Following prior work [42], we also adopt a HAN as the network structure of N^T_{loc} to process self-reported location. Specifically, we first encode X^T_{loc} into embeddings using a word-embedding encoder and then a GRU layer followed by an attention layer to generate the final representations. This is identical to the tweet text component.

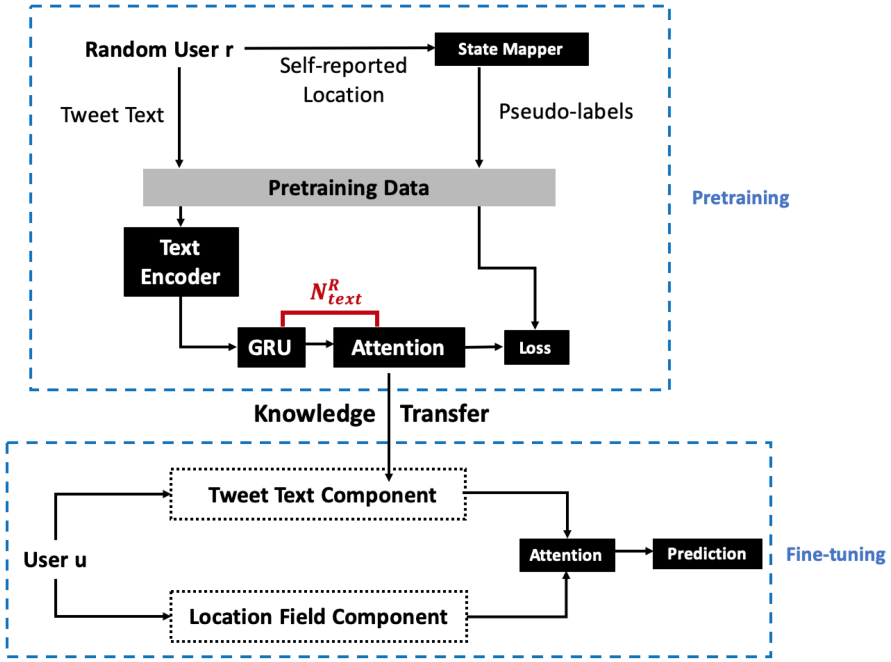


Fig. 3. Overview of the related concept pretraining.

3.2.3 Feature Fusion. Since we do not know whether the tweet text or the location field is more important for the location inference task, we incorporate an additional attention layer to combine both pieces of information. In this way, different weights are assigned to the two components, depending on their importance for the prediction task. We then include a fully connected layer before returning the final prediction.

3.3 Related Concept Pretraining

Our first improvement is to integrate relevant related location concepts into the model training. This approach can be viewed as a way to enrich the feature space.

3.3.1 Model Background. More formally, suppose we have a dataset with random users. The user set is represented as $U^R = \{u_1^R, u_2^R, \ldots, u_n^R\}$, where n is the total number of users and u_j^R is the jth user. Each user u_j^R contains tweet text and location field features, like users in U^T. Similarly, let $X_{text}^R = \{x_{1,text}^R, x_{2,text}^R, \ldots, x_{n,text}^R\}$ be the tweet text features of users U^R and $X_{loc}^R = \{x_{1,loc}^R, x_{2,loc}^R, \ldots, x_{n,loc}^R\}$ be the self-declared location features.

3.3.2 Model Design.

Model Overview. Figure 3 shows the overview of the Related Concept Pretraining approach. It consists of two steps: pretraining and fine-tuning. First, we pretrain a model N_{text}^R, which takes tweet text X_{text}^R as input. The labels are then built using a self-supervised signal. These labels are state labels and are derived from the location field X_{loc}^R. We refer to these new labels as *pseudo-labels*, since these are extracted using self-supervised learning. (We will discuss our decision to select the location field information as the self-supervised signal in the next subsection.) The difference between pseudo-labels and ground truth labels is that pseudo-labels are extracted automatically without human annotation, while ground truth labels are manually annotated by humans.

Figure 3 also shows a simplified version of the base model (see Figure 2 for more detail). Specifically, N^R_{text} will be used within the tweet text encoder in the tweet text component of the base model, while we keep other components unchanged. Finally, we make the city prediction.

Model Structure. For pretraining, we choose the location field address as the self-supervised signal for the following reasons: (1) Self-declared location information can be an informative feature that contributes to model accuracy [30]. While it is not always the user's actual address, it can still guide the model, serving as a reasonable estimate. (2) Self-reported location is shared publicly by many users on Twitter/X [12], making it straightforward to collect.

Because self-declared locations are often noisy, i.e., they could be a city, a state, or even invalid, we implement a function to map users' self-reported location into states. We build a city-to-state dictionary. We then created a mapper function that searches for a city name (and/or a state name) in the user profile and returns the corresponding state mapping using the city-to-state dictionary. We choose states as the signal because of their hierarchical relationship to cities and because state information is more commonly found in the location field than city information [12]. More formally, we first use a state mapper function *mapper* to convert the self-report location into the corresponding state in the United States:

$$Y^R = \{y^R_j\} = \{mapper(x^R_{j,loc})\}, j \in [1, n].$$

For example, if the value of $x^R_{j,loc}$ is "SF, Cali," then we convert it into the state "California." Then, we apply a filter function $filter$, which returns false if the location field information can not be mapped into a state. Those users that we cannot map to a state are filtered out:

$$Y'^R = \{y^R_j \subset Y^R | filter(y^R_j) = True\}, j \in [1, n]$$
$$X'^R_{text} = \{x^R_{j,text} \subset X^R_{text} | filter(y^R_j) = True\}, j \in [1, n].$$

Finally, we pretrain a model with $X'^R_{text} = \{x'^R_{1,text}, x'^R_{2,text}, \ldots, x'^R_{n',text}\}$ as input and $Y'^R = \{y'^R_1, y'^R_2, \ldots, y'^R_{n'}\}$ as the class labels, where n' is the number of remaining users. Similar to the *Tweet Text Component* of the base model, we use a HAN as the structure for the neural model N^R_{text}.

Fine-tuning. For fine-tuning, we replace the original encoder N^T_{text} with N^R_{text}. We still use N^T_{loc} to process the location field and an attention layer to combine the information from the tweet text and the location field. We hypothesize that this external knowledge will improve the predictive accuracy of the location inference model, because the pretraining using state will result in a user embedding that is *closer* to the city task.

3.4 Data Augmentation Based on External Resources

Intuitively, our other strategy is to increase the number of users in the training dataset, U^T. We accomplish this by taking location-related articles, e.g., Wikipedia, and segmenting the information into different synthetic users.[3]

3.4.1 Model Background. More formally, suppose we have a set of articles $A = \{a_1, a_2, \ldots, a_g\}$, where g is the total number of articles and a_k is the kth article. Each article a_k contains a title e_k and content c_k, which consists of $|c_k|$ sentences.

3.4.2 Model Design.

Model Overview. Figure 4 presents the overview of the proposed *Data Augmentation Based on External Resources*. We first download online articles related to cities. Each article includes a title e and content c, which is segmented into different fragments. We then pair each fragment with the

[3]The segment function is based on sentence segmentation, with each synthetic user containing 20 sentences.

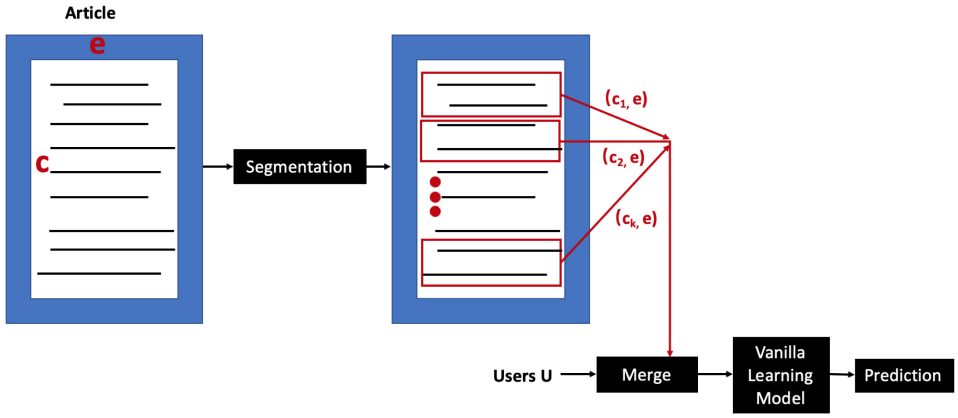


Fig. 4. Overview of data augmentation based on external resources.

title and create synthetic users. Next, we combine all the synthetic users from all the articles with users U^T from the training dataset. Finally, we use our base model to train the combined dataset and make predictions.

Synthetic User Generation. Our synthetic user generation approach focuses on increasing the size of the training data by incorporating knowledge from external sources. In particular, we hypothesize that utilizing location-related information available online can improve the model's performance, since the labeled training data may not contain sufficient information for model prediction. Recall that some cities have very few labels. We leverage the effectiveness of Wikipedia in enhancing location classification [34] and utilize it as our source of external knowledge. Specifically, as Figure 4 shows, we download the Wikipedia articles related to different cities that exist in the training dataset. For each article, we use a segment function *segment* to divide the page content into fragments and assign the same label (article title) to each fragment. This yields:

$$U^W = \{segment(c_k, e_k)\}, k \in [1, g].$$

We then combine the newly generated users U^W with the users U^T from the training set. Finally, we train a model with the base model structure using the combined dataset and make predictions. Note that, for this method, we do not conduct fine-tuning. Instead, we only add the new training examples to the existing training dataset and train the model from scratch.

3.5 Hybrid Approach

In our work, we also investigate the performance of merging the pretraining and augmentation methods, leveraging the respective advantages of both. Specifically, we first pretrain a model N^R_{text}, as the same process in related concept pretraining. Then, using the combined dataset $U^W + U^T$ generated from the proposed data-augmentation approach, we adopt the base model structure with N^R_{text} as the tweet text encoder to train and make final predictions.

4 Analysis of Location Datasets

In this section, we first introduce the datasets used in our empirical evaluation, including the training sets, the pretraining set, and the Wikipedia dataset. Then, we discuss the reliability of different labeling methods for determining the ground truth location. Finally, we use this reliability analysis to suggest a method for labeling location datasets and apply the method to label the dataset

we collected. For the remaining datasets, we keep the existing labels to ensure consistency with prior research.

4.1 Datasets

TwitterUS. The first dataset we use is a publicly available dataset, TwitterUS [50], which consists of 429K users in the training set, 10K users for development, and 10K test users in a North American region. The ground truth location of a user is set to the first geotag of the user in the dataset. Following prior work [30], we assign the closest city to each user's ground truth location using the city category approach designed by Han and colleagues [23]. Because our focus is a low-resource setting, i.e., a small number of labeled examples, we extract 50 random users per category (city). We exclude cities if there are fewer than three users associated with them, leaving a dataset containing 18,470 users.

GeoText. GeoText is a dataset from Eisenstein et al. [18] that contains 9,500 users with geographical coordinates for each user. All users come from the contiguous United States. The ground truth location of a user is also set to the first geotag of the user in the dataset. Similar to the TwitterUS dataset, we apply a filtering criterion of a minimum of three users per category, resulting in a total of 9,373 remaining users. Because handles are not shared due to privacy concerns, we only use text for this dataset.

GeoDecahoseSample. We collect a dataset that consists of approximately 10,000 users sampled from the Twitter Decahose (a 10% sample of tweets). Each user has at least one geotagged tweet in the United States. We use the geotagged tweet and the profile to determine the location of the user. Similar to other datasets, we remove cities that do not meet the minimum requirement of having at least three users. We call this dataset GeoDecahoseSample, resulting in a final dataset containing 6,559 users.

GeoState. For our data-collection process, we randomly selected a consecutive two-week period in each of the quarters in 2021–2022 (last two quarters for 2021 and first two quarters for 2022) from the Twitter Decahose [56]. We then collected the profile information of users who posted tweets and identified their locations using their profiles. Next, we randomly selected up to 1,000 handles from each state, removing suspended accounts or accounts that did not post any tweets. This results in a dataset containing approximately 900 users per state.

Wikipedia. We first download all the articles associated with American cities that also exist in our training datasets from Wikipedia. Next, we extract articles specifically from the "main article" section of each city, which includes a link that redirects the reader to another Wikipedia article containing more in-depth information on a specific topic related to the main topic of the current page. For example, we retrieve the Wikipedia article for the city of Boston, as well as the article for the history of Boston from the "main article" section. Using this approach, we generate approximately 5,100 synthetic users.

4.2 Location Labeling Reliability

There are primarily three methods for labeling users in location datasets: using self-reported location, using the first geotagged location, or using a majority vote of geotagged locations. For example, Roller et al. [50] set the location of the first geotagged post of a user in the dataset as the ground truth location. Han and colleagues [25] propose using a majority vote method. They only keep users with at least 10 geotagged tweets to get a reliable estimate of a user's primary location. To be eligible as a data record in the geotagged dataset, a user must have 50% of their tweets coming from the same area. Vijayaraghavan et al. [57] use the self-reported location as the ground truth.

Table 1. Match Rate for Different Methods during Different Time Periods

	Most Recent Two Years		Most Recent Three Years	
	GeoDecahoseSample	GeoState	GeoDecahoseSample	GeoState
Segments	0.888	0.895	0.818	0.803
Segments vs. Majority Vote	0.874	0.889	0.816	0.802
Segments vs. Self-reported Location	0.613	0.533	0.6	0.523
Segments vs. First GeoDecahoseSample Tweet	0.662	0.657	0.618	0.588

To test the validity of each of these labeling methods, we first segment the geotagged tweets of each user into distinct time periods and compare the matching rates of these segments with a yearly granularity. To obtain a reliable estimate, we follow the rule of Han and colleagues [25]: For each segment, a user must have at least 10 geotagged tweets, and 50% of their tweets must originate from the same area. We remove those that do not satisfy the condition. This method aims to identify whether users change their primary location over time. Then, we perform a comparison between the labels determined by the tweet segments and the location labels from the first geotagged method, the majority vote method, and the self-declared location method, respectively. We consider a match to occur only when (1) the segment locations are the same during a one-year time window, and (2) the segment locations match the location determined by the specific labeling method.

Because TwitterUS and GeoText only provide the coordinates of the first tweet, we choose GeoDecahoseSample and GeoState for this analysis and use their geotagged tweets. Table 1 presents the match rate of different methods for total time windows of two years and three years, respectively. For the GeoDecahoseSample dataset, when comparing two consecutive segments, the match rate is 0.888, indicating that users who share their geotagged information tend to stay in the same location for an extended period. The majority vote method also has a high match rate (0.874), suggesting that it is a relatively reliable labeling technique even though the match rate is slightly lower than the segment method. Manual inspection of tweets reveals that the decrease is mainly due to the impact of geotagged locations from previous time periods when a user may have been in a different location (note that the majority vote method involves considering all of a user's tweets across all times for labeling purposes). The profile location method yields a match rate of 0.613. Though not as accurate as majority vote, it still provides insights into why **Related Concept Pretraining (RCP)** may help with city-level prediction. Finally, the match rate for the first geotagged location method is 0.657. This approach is better than using profile location, but not as strong as segments or majority vote. For GeoState, we see a similar trend. Both the segment method and majority vote have a high match rate, and the results are comparable to the GeoDecahoseSample dataset. Similarly, the match rate for the first geotagged location method is significantly lower, indicating that it may not be an effective labeling approach.

For the three-year window, we see that the match rate is lower than the match rate for the two-year window by 1% and 7%. However, the segment method and majority vote still yield favorable results with match rates exceeding 0.8.

4.3 Our Ground Truth Labeling Approach

The labeling approach we recommend is a variant of the majority vote method. We choose not to use the segment method, since it assumes that users have multiple years of geotagged tweets. We begin by filtering users who do not have a minimum of 10 geotagged tweets with a valid location. Then, we only consider a user's most recent 30 tweets for the majority vote calculation. This helps mitigate the influence of older tweets.[4] This process results in a final dataset of 6,559 users for

[4]The requirement of 30 geotagged tweets was determined empirically.

the GeoDecahoseSample dataset. For the other two datasets (TwitterUS and GeoText), we use the ground truth provided, since we do not have the historical data.

5 Empirical Evaluation

In this section, we evaluate our proposed models. We first describe the experimental setup (Section 5.1) and baseline models (Section 5.2). Then, we conduct a detailed empirical evaluation to assess the effectiveness of our approaches for location prediction with limited training data (Section 5.3).

5.1 Experimental Settings

In our experiments, we use one NVIDIA Tesla P4 GPU on a Google Cloud compute engine with 60 GB of memory. We initialize word embeddings with GloVe [45] and encode tweet text using CLIP [47]. We employ the Adam update rule [33] to optimize our model. Weight, bias, and context vector are randomly initialized for the attention layers and then normalized to have a mean value of 0 and a standard deviation of 0.05. A batch size of 32 is used for training, and tweets per user are limited to a maximum of 200, with shorter sequences padded. We set the learning rate as 0.0001. We use a bidirectional GRU with hidden size being 512 and layer being 1.[5] For users, including synthetic users, who lack location field information, padding is also utilized. We collect the Wikipedia content of 342 cities and generate appropriately 5,100 synthetic users. We keep the test set unchanged and run each experiment five times with a random seed. For the proposed data-augmentation model and the hybrid model, each synthetic user takes 20 sentences as that user's tweets (see Section 5.7 for the sensitivity analysis). New data are incorporated into the training and validation sets, while the test set is also kept unchanged and only used for assessing the model's performance. We report the average results of the following commonly used metrics:

— **Accuracy** The percentage of correctly predicted cities.
— **Acc@161** The percentage of predicted cities that are within a 161 km (100 miles) radius of true locations. This metric is used to capture near-misses.
— **Median** The median distance computed from the distances between the predicted cities and the corresponding actual location coordinates.

5.2 Baseline Models

We consider seven different models, which include the state-of-the-art models for location inference from previous works.

— **Support Vector Machine (SVM)** has been successfully used for the location inference task using Twitter/X data [39].
— **Naive Bayes (NB)** has also been successfully used for location detection [23].
— **Word Attention** Minura et al. [42] propose an attention-based model for this task that combines text, metadata, and network data. Specifically, they map all these components into an embedding space and then use an attention layer to connect them. We use a variant of their model that combines text and metadata.
— **Base Model** We use the base learning model presented in Section 3.2 as another baseline.
— **Transformer** We use the transformer model as another baseline. Specifically, we first convert tweet text into embeddings and then use a transformer architecture for our learning task.

[5]For detailed implementation, please refer to our source code.

— **Traditional Augmentation** We randomly apply the augmentation techniques in EDA [58] to tweet text and use the base model structure for the experiments. The augmentation techniques include synonym replacement, random word insertion, random word swap, and random word deletion.
— **Hierarchical Location Prediction Neural Network (HLPNN) [30]** Huang and Carley [30] introduce a hierarchical model where they use character-level and word-level embeddings for this task with state information guiding the training of the city prediction. Specifically, each tweet is represented as both word embeddings and character embeddings. Then, a transformer is used to connect all the components. Finally, the optimization goal is to minimize the loss of both state prediction and city prediction. We employ a variation in which we utilize CLIP to encode tweet text, since it has shown significant improvement in demographic inference on Twitter/X [37].
— **Emoji Model** This method is proposed by Liu and Singh [36]. The base model is similar to ours, but this approach also incorporates emojis as an independent component to further improve the performance. In the original paper, the authors specify that emojis should be sorted based on their posting time, similar to tweets. However, since the available public datasets are from earlier periods, we only have the timestamp information for the GeoDecahoseSample dataset. Therefore, we use that dataset for these results.
— **Traditional Transfer Learning (TL)** This transfer learning method proposed by Liu and colleagues [37] works by pretraining a model on a larger labeled dataset and then fine-tuning it using the smaller training set. In our work, we adopt a similar approach by conducting model pretraining using one of the three available datasets and then fine-tuning the pretrained model using a different dataset. Specifically, we pretrain a model using TwitterUS and apply the model to the GeoDecahoseSample dataset for fine-tuning and make predictions for GeoDecahoseSample.

5.3 Experimental Results

Results of Accuracy&Acc@161. Table 2 shows the results of the state-of-the-art and the proposed models. For TwitterUS, when using text only, we see that the classic machine learning models perform well in terms of accuracy. SVM gets higher accuracy than both the word-embedding model and the base model. The accuracy of the base learning model is higher than the word-embedding model, indicating the effectiveness of sentence-level embedding. The transformer and the baseline model with RNN+Attention perform similarly. TL performs the best among the state-of-the-art, with an accuracy of 0.098. The proposed pretraining approach achieves a comparable accuracy to that of TL, with a difference of 0.1%, and it is 1.8% higher than the base model. The proposed data-augmentation method achieves an accuracy of 0.119, which is 2.1% higher than TL. The hybrid model has the highest accuracy, with an improvement of 2.3% over TL. For Acc@161, we see that classic models do not perform as well. TL is again the best performer among all state-of-the-art models, and the hybrid model achieves the highest result with an improvement of 2.2% over TL.

For GeoDecahoseSample, the traditional augmentation-based model achieves the highest accuracy compared to other state-of-the-art models. It is 0.1% lower than the proposed pretraining method, which has an accuracy of 0.478. The proposed data-augmentation approach has the highest result, with a 4.2% improvement over the best state-of-the-art model. For Acc@161, the hybrid model performs the best. It is 6.1% higher than the traditional augmentation-based method.

For GeoText, a similar trend is observed as in the other datasets. TL achieves the best result among all the state-of-the-art models, and the proposed pretraining approach has comparable accuracy to TL, with only a small difference of 0.4%. The hybrid model performs the best and is 1.6%

Table 2. Comparisons of Our Methods and the State-of-the-art on Different Datasets

	TwitterUS		GeoDecahoseSample		GeoText	
	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161
Text						
NB	0.061	0.158	0.313	0.418	0.103	0.336
SVM	0.087	0.191	0.423	0.531	0.099	0.334
Word Attention	0.035	0.11	0.312	0.432	0.088	0.284
Base Model	0.081	0.294	0.455	0.655	0.114	0.35
Emoji Model	NA	NA	0.453	0.661	NA	NA
Transformer	0.079	0.301	0.436	0.645	0.121	0.383
Traditional Augmentation	0.078	0.279	0.477	0.685	0.118	0.362
TL	0.098	0.329	0.464	0.684	0.146	0.409
HLPNN	0.091	0.309	0.475	0.683	0.132	0.399
RCP (our model)	0.099	0.331	0.478	0.71	0.142	0.423
DAER (our model)	0.119	0.341	**0.519**	0.727	0.154	0.436
Hybrid (our model)	**0.121**	**0.351**	0.516	**0.746**	**0.162**	**0.451**
Text+Loc						
NB	0.055	0.153	0.306	0.42		
SVM	0.094	0.199	0.392	0.501		
Word Attention	0.084	0.226	0.445	0.616		
Base Model	0.111	0.343	0.465	0.671		
Emoji Model	NA	NA	0.444	0.653	NA	
Transformer	0.1	0.268	0.466	0.661		
Traditional Augmentation	0.113	0.349	0.488	0.704		
TL	0.117	0.363	0.505	0.727		
HLPNN	0.113	0.371	0.505	0.729		
RCP (our model)	0.114	0.357	0.48	0.709		
DAER (our model)	0.124	0.363	**0.525**	0.74		
Hybrid (our model)	**0.128**	**0.377**	0.518	**0.746**		

Related Concept Pretraining is denoted by RCP and Data Augmentation based on External Resources is denoted as DAER. We report results under two different feature settings: Text and Text+self-reported location. "N/A" signifies no applicable results for the given dataset.

higher than TL. Also, the hybrid model outperforms the state-of-the-art in terms of Acc@161, with an improvement of 4.2%. Overall, we see that the proposed pretraining method generally achieves comparable results to the best state-of-the-art model without using any externally labeled dataset. The proposed augmentation model has higher results than all the state-of-the-art models, and the hybrid model usually performs the best.

When using both text and self-reported location, for TwitterUS, we see that word-embedding model still performs worse than the base model, but both perform better than the classic machine learning models. The transformer model again has comparable accuracy to the baseline model with RNN+Attention. The proposed pretraining method has comparable accuracy to other state-of-the-art models, such as TL and HLPNN. The proposed augmentation method performs better than all the state-of-the-art models, and the hybrid model achieves the highest accuracy, which is 1.1% higher than TL, the best state-of-the-art model. Additionally, the hybrid method performs best for Acc@161. For GeoDecahoseSample, the proposed data-augmentation model generally achieves the best results. Overall, we have a 2% improvement in accuracy over the best state-of-the-art models, including TL and HLPNN, and a 1.7% improvement for Acc@161.

Table 3. Comparisons of Our Methods and the State-of-the-art on Different Datasets for Median Error Distance

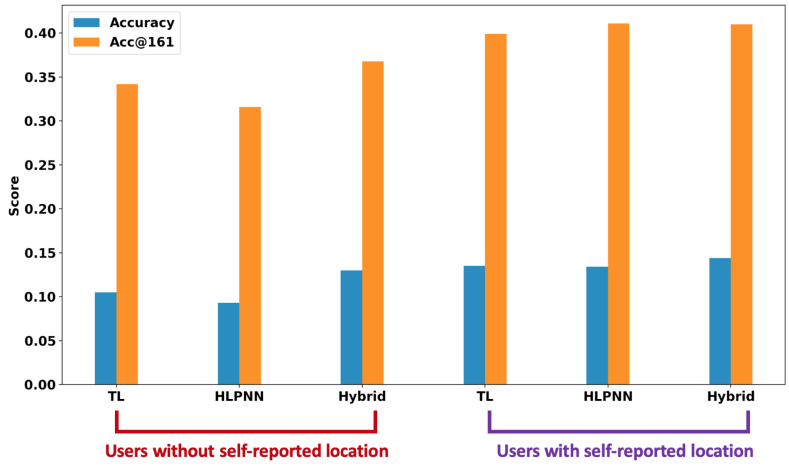
	TwitterUS	GeoDecahoseSample	GeoText
Text			
NB	968.7	427.0	620.2
SVM	1,018.3	85.9	650.1
Word Attention	1,062.	385.8	1,519.7
Base Model	373.4	27.2	561.4
Emoji Model	NA	27.4	NA
Transformer	356	38.3	446.1
Traditional Augmentation	403.6	20.1	525.9
TL	340.7	23.4	347.3
HLPNN	357.2	20.7	351.6
RCP (our model)	332.4	17.7	318.2
DAER (our model)	330.2	1.7	306.5
Hybrid (our model)	**317.5**	**0**	**265.6**
Text+Loc			
NB	1,253.8	427.0	
SVM	995.6	161.6	
Word Attention	793.1	30.8	
Base Model	305.1	23.5	
Emoji Model	NA	29.9	N/A
Transformer	591.4	21.3	
Traditional Augmentation	310.6	12.3	
TL	285.2	3.4	
HLPNN	**272.3**	3.8	
RCP (our model)	284.2	15.7	
DAER (our model)	296.0	**0.0**	
Hybrid (our model)	274.3	2.5	

Related Concept Pretraining is denoted by RCP and Data Augmentation based on External Resources is represented as DAER. "N/A" signifies no applicable results for the given dataset.

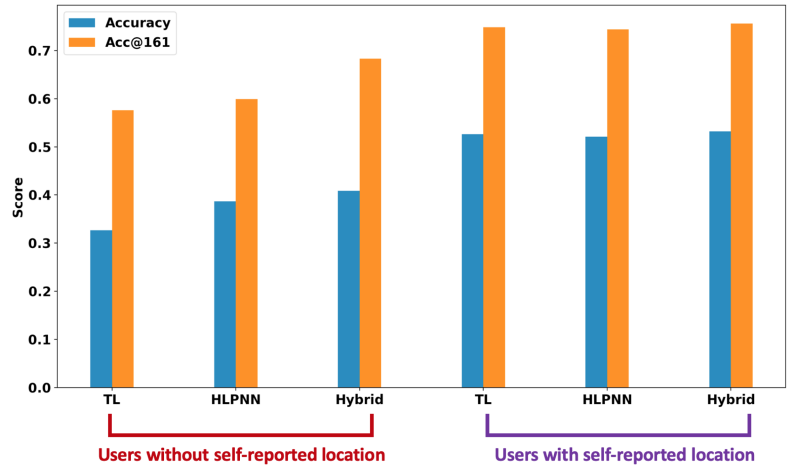
Results of Median Error Distance. Table 3 displays the median distance of the state-of-the-art models and the proposed models. For TwitterUS, the error distance is reduced by approximately 23.2 kilometers when comparing TL to the best proposed model. We see a similar trend for the GeoDecahoseSample dataset. It is worth noting that the median error distance of our best proposed model is 0, showing that our method greatly reduces the error distance. For GeoText, the hybrid model has an error distance that is 39.2 kilometers smaller than the state-of-the-art models. When using both text and self-declared location, our model generally performs similarly or better than the state-of-the-art models.

5.4 Evaluation of Users with/without Self-reported Location

Cheng and colleagues [13] found that only 26% of Twitter/X users provide location information at the city level in their profiles. In our study of the TwitterUS dataset, we found that 42% of users have self-reported city information provided. When this is available, it is useful to use. However, because many users do not provide this information, inferring location is important. In this section, we present the results of our models for both types of users. For users without locations in their profiles, we report the performance of our model using only text and compare it to the



(a) GeoText



(b) GeoDecahoseSample

Fig. 5. Results, including accuracy and Acc@161 for users who self-reported their location and those who did not. For each subfigure, the X axis shows the different types of approaches (TL, HLPNN, and the hybrid model) and the Y axis is accuracy or Acc@161.

state-of-the-art models. For users with location information, we show the results using both the text and the location field information. Similar to the previous section, this analysis includes the best state-of-the-art models and the hybrid model.

Figure 5 presents the comparisons between our model and the best state-of-the-art models for accuracy and Acc@161 of the TwitterUS dataset and the GeoDecahoseSample dataset. For TwitterUS, our model achieves better results for both groups of users in terms of accuracy. The evaluation based on Acc@161 shows that the proposed model performs better than or comparably to the best state-of-the-art model. For GeoDecahoseSample, our model generally performs better, with a marginal improvement when using both text and self-reported location. These results further demonstrate the ability of our proposed model to attain a higher accuracy than other models for users who share their city-level location and those who do not.

Table 4. Number of Users for Each City from the Sampled Datasets

	TwitterUS	GeoDecahoseSample	GeoText
5 Users per City	1,894	1,186	1,214
10 Users per City	3,764	1,833	1,995
20 Users per City	7,459	2,485	3,020

We consider three different sample sizes of users per city—5, 10, and 20.

Table 5. Evaluation on Sampled Datasets of Best Models (TL, HLPNN, and the Hybrid Model)

	TwitterUS						GeoTagged						GeoText					
	5 per City		10 per City		20 per City		5 per City		10 per City		20 per City		5 per City		10 per City		20 per City	
	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161	Acc	Acc@161
TL	0.049	0.216	0.066	0.279	0.094	0.328	**0.285**	0.611	0.341	0.651	0.412	0.692	0.031	0.237	0.043	0.277	0.059	0.312
HLPNN	0.049	0.208	0.069	0.277	0.098	0.337	0.176	0.506	0.271	0.595	0.391	0.664	0.019	0.185	0.026	0.22	0.041	0.271
Hybrid	**0.078**	**0.283**	**0.099**	0.321	**0.113**	**0.358**	**0.285**	**0.636**	**0.357**	**0.663**	**0.43**	**0.702**	**0.051**	**0.285**	**0.063**	**0.337**	**0.071**	**0.334**
Improvement	**2.9%**	**6.7%**	**3%**	**4.2%**	**1.5%**	**2.1%**	0%	**2.5%**	**1.6%**	**1.2%**	**1.8%**	**1%**	**2%**	**4.8%**	**2%**	**6%**	**1.2%**	**2.2%**

In the last row, we show the improvement of our model over the state-of-the-art models.

5.5 Evaluation of Smaller Training Datasets

To evaluate the performance of our models in real-world scenarios, where each city label is associated with only dozens or fewer samples, we select three samples from each dataset as the training data. Each sample consists of a different number of users per city: up to 5 users per city, 10 users per city, and 20 users per city. Table 4 displays the number of training samples for each sample.

Table 5 presents the comparisons of results when using samples with different numbers of users for training. For ease of exposition, we only compare the best state-of-the-art models (TL and HLPNN) and the hybrid model. We use both text and self-reported location when training our model when using the GeoDecahoseSample and TwitterUS datasets and text only when using the GeoText dataset, since this dataset does not include the location field information. We focus on accuracy and Acc@161.[6] From the table, we see that for sampled TwitterUS, when using 5 users per city, our model achieves the best results, with an improvement of 2.9% for accuracy and 6.7% for Acc@161. The difference between the TL and the HLPNN is marginal. When considering 10 users per city, our model still performs 3.3% and 4.2% better than the previous state-of-the-art for accuracy and Acc@161, respectively. For the case with 20 users per city, we observe a similar trend. For the GeoDecahoseSample dataset, we see that the hybrid model generally performs significantly better than the TL and the HLPNN models. The performance of the HLPNN model is much worse than the TL and the hybrid models. We posit that this is due to the relatively small size of the sampled dataset. For GeoText, the hybrid model again performs the best, and both the TL and the hybrid models have better results than the HLPNN model. Figure 6 summarizes these results using a heatmap. The y-axis shows the different methods and the x-axis shows the number of users per city for three different datasets. While all the Acc@161 are more accurate for the GeoTagged dataset, the Hybrid method is always better than or just as good as the other methods when the number of users is limited.

5.6 Evaluation on Fewer Cities

For some applications, researchers limit their studies to a small number of target cities. For this analysis, we reduce the number of cities of interest based on population. The first sample consists of the 10 cities with the highest population, and the second sample consists of the 10 cities with the lowest population, based on data from the United States Census Bureau.[7] Figure 7 shows the

[6]Note that the median distance metric is also applicable.

[7]The list can be accessed through https://www.biggestuscities.com/

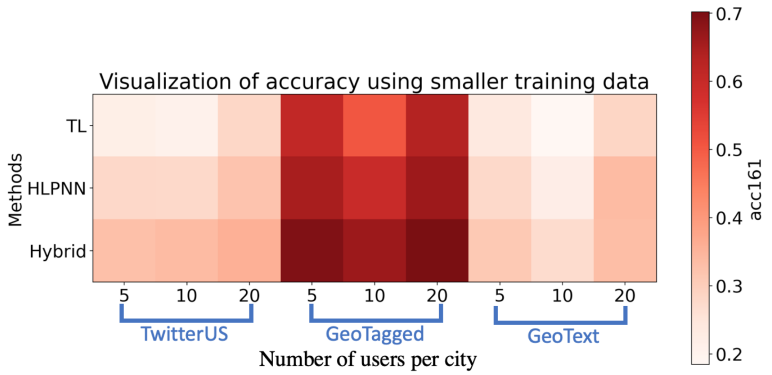


Fig. 6. Visualization of the Acc@161 comparing different methods using limited data.

cities and number of users in our dataset that are located in each city.[8] It is not surprising to see that the most populated cities tend to have more labeled data, and the least populated cities have less. Due to the significant distance between these cities, we use accuracy and F1 score rather than Acc@161 and median distance to compare the performance of the models.

Table 6 shows the results for GeoDecahoseSample and GeoText. We focus on comparing the performance of the hybrid model with that of the best performing state-of-the-art models: TL and HLPNN. For the GeoText dataset, the TL and the hybrid models outperform the HLPNN model for both groups of cities. While the accuracy difference between the TL model and the proposed model is insignificant, we see that the F1 score of the hybrid model is much higher than the TL model, with an improvement of 5.9% for the most populated cities and 8.3% for the least populated cities, again demonstrating its ability to accurately identify cities with very different characteristics and limited data. For the GeoDecahoseSample dataset, we see a similar trend. The accuracy of the three models for the most populated cities is comparable, but the F1 score of our model is 3.1% higher than the best baseline. For the least populated cities, our model significantly outperforms the baseline models, with an improvement of 5% in accuracy and 7.2% in F1 score. These results suggest that our model can be deployed for studies containing different numbers of cities, as well as different population characteristics.

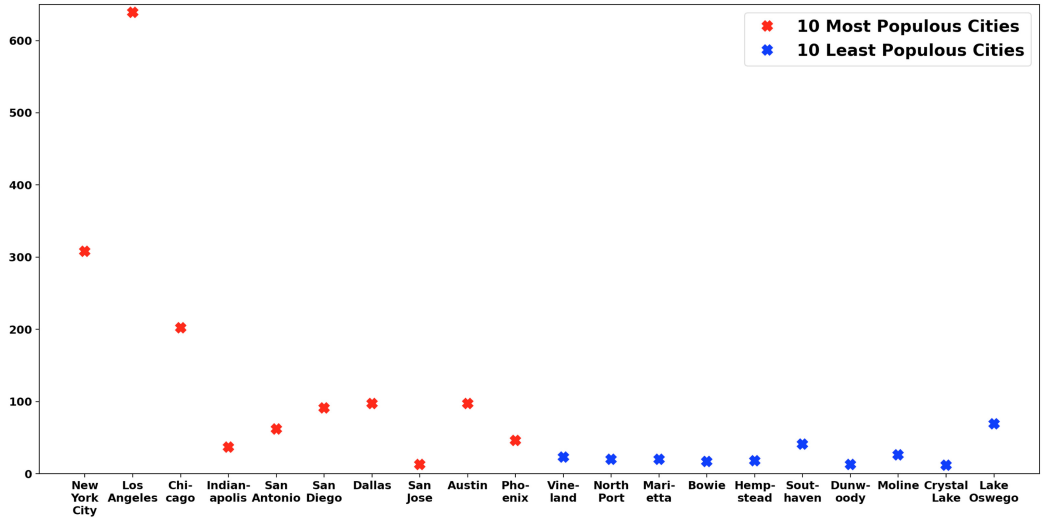
5.7 Sensitivity Analysis of Wikipedia

In this section, we study how the model performance changes when different numbers of sentences from Wikipedia pages are used to represent each synthetic user. Table 7 shows the results of assigning 20, 30, or 50 sentences to each synthetic user using the proposed data-augmentation model. As we can see, in terms of both accuracy and Acc@161, different methods perform best for each dataset. However, the difference between the lowest performer and highest performer is within 1% across all the datasets, indicating that the specific selection of sentences has minimal impact.

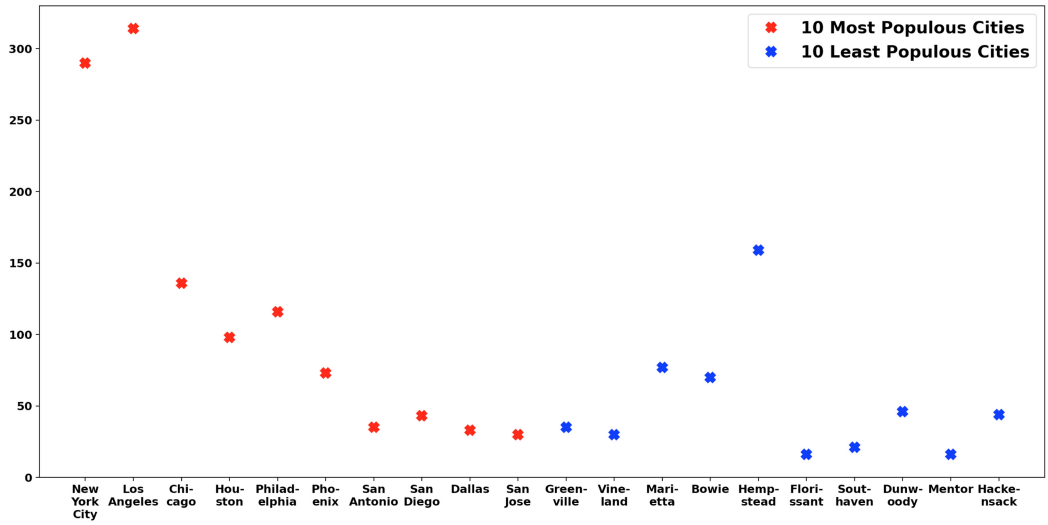
5.8 Model Efficiency

In general, each experiment takes between 5 and 10 hours to run. For RCP, the pretraining phase extends the overall run time by appropriately 5 hours, and it is only necessary to do once. Its

[8]Note that, when selecting cities for each dataset, we only consider cities that are present within that specific dataset, which accounts for the variation in cities between the two datasets.



(a) GeoText



(b) GeoDecahoseSample

Fig. 7. Number of users for the 10 most populous cities and least populous cities.

Table 6. Results of the Proposed Model and Baselines for 10 Most Populated Cities in the US and 10 Least Populated Cities in the US

	GeoText				GeoDecahoseSample			
	Most Populous		Least Populous		Most Populous		Least Populous	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TL	0.536	0.339	0.396	0.204	0.858	0.725	0.715	0.602
HLPNN	0.455	0.212	0.32	0.112	0.863	0.727	0.531	0.366
Hybrid	**0.547**	**0.398**	**0.4**	**0.287**	**0.864**	**0.758**	**0.765**	**0.674**

Table 7. Comparisons of Different Segmenting Methods that Involve Using Different Numbers of Sentences from Corresponding Wikipedia Articles as Tweets when Creating Synthetic Users

	GeoDecahoseSample		TwitterUS		GeoText	
	Accuracy	Acc@161	Accuracy	Acc@161	Accuracy	Acc@161
20 Per User	**0.519**	**0.727**	0.119	0.341	0.154	0.436
30 Per User	0.516	0.714	**0.123**	**0.348**	0.147	0.431
50 Per User	0.514	0.717	0.114	0.338	**0.156**	**0.437**

fine-tuning phase is cost-effective compared to other methods, as it converges rapidly. With DAER, the larger training data causes a longer duration compared to the base model (appropriately 5 hours). For the hybrid model, it still converges quickly because of the pretrained model. In general, our proposed methods have a comparable runtime to the other models.

6 Conclusions and Future Work

In this article, we present different approaches for city-level inference when limited training data are available. This setting is particularly important when conducting a study with a large number of cities because of the cost associated with labeling training data. The first method, Related Concept Pretraining, leverages a self-supervised learning framework to enhance user representations. Our experiments demonstrate the effectiveness of this approach, particularly for predicting users based on text data alone. Our second method, Data Augmentation Based on External Resources, utilizes external knowledge to increase the number of training examples by adding synthetic users. Empirical results show that both of these methods perform better than the state-of-the-art for location inference. We also find that combining these two approaches into a hybrid model that leverages their respective advantages to enhance user representations and expand the size of the training data further improves the accuracy of our model. This is true irrespective of the available features.

Our work also includes two sensitivity analyses. First, we restrict the number of training examples available for each label (city). Using these subsamples, we compare the performance of our best model against the best state-of-the-art models. The results show that our proposed model generally outperforms the other models, indicating its ability to perform well even when the number of training examples is limited. We also consider focusing on a smaller number of cities within a single analysis, specifically narrowing down the classification task to only 10 cities based on population. In this setting, our model again shows significant improvements, suggesting its potential utility for tasks involving different numbers of cities.

Finally, we examine various labeling methods for determining ground truth cities using two different datasets. We find that the majority vote method is a more reliable strategy, compared to assigning the location based on the first geotagged entry or relying on self-declared location. We propose a labeling method that is based on majority vote, potentially addressing the limitations of prior methods.

There are a number of possible future directions. First, finding new sources of external knowledge may further enhance the model performance. One potential approach is to utilize chatbots like ChatGPT, which has shown great text generation capabilities and can introduce a different but relevant type of location information into our model. Additionally, while our article primarily focuses on city prediction within North America, we would like to extend the applicability of our models to a global dataset containing a broader range of cities. Other possible directions include considering multi-lingual location detection and testing this approach on other social media platforms.

References

[1] S. Agrawal and A. Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *ECIR*. Springer.

[2] F. Al Zamal, W. Liu, and D. Ruths. 2012. Homophily and latent attribute inference: Inferring latent attributes of Twitter users from neighbors. In *ICWSM*.

[3] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. 2004. Web-a-where: Geotagging web content. In *SIGIR*.

[4] M. Berggren, J. Karlgren, R. Östling, and M. Parkvall. 2016. Inferring the location of authors from words in their texts. *arXiv preprint arXiv:1612.06671* (2016).

[5] F. Bilhaut, T. Charnois, P. Enjalbert, and Y. Mathet. 2003. Geographic reference analysis for geographic document querying. In *HLT-NAACL Workshop on Analysis of Geographic References*.

[6] L. Bode, P. Davis-Kean, L. Singh, T. Berger-Wolf, C. Budak, G. Chi, A. Guess, J. Hill, A. Hughes, J. Jensen, et al. 2020. Study designs for quantitative social science research using social media. *PsyArXiv* (2020).

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Computat. Ling.* 5 (2017), 135–146.

[8] O. Buyukokkten, J. Cho, H. Garcia-Molina, L. Gravano, and N. Shivakumar. 1999. Exploiting geographical location information of web pages. In *WebDB* (1999).

[9] K. Carley, M. Malik, P. Landwehr, J. Pfeffer, and M. Kowalchuck. 2016. Crowd sourcing disaster management: The complex nature of Twitter usage in Padang Indonesia. *Safety Sci.* 90 (2016), 48–61.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

[11] X. Chen, Y. Wang, E. Agichtein, and F. Wang. 2015. A comparative study of demographic attribute inference in Twitter. In *ICWSM*.

[12] Z. Cheng, J. Caverlee, and K. Lee. 2010. You are where you tweet: A content-based approach to geo-locating Twitter users. In *CIKM*.

[13] Z. Cheng, J. Caverlee, and K. Lee. 2013. A content-driven framework for geolocating microblog users. *ACM Trans. Intell. Syst. Technol.* 4, 1 (2013), 1–27.

[14] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[15] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[16] A. Culotta, N. Kumar, and J. Cutler. 2015. Predicting the demographics of Twitter users from website traffic data. In *AAAI*.

[17] M. Ebrahimi, E. ShafieiBavani, R. Wong, and F. Chen. 2018. A unified neural network model for geolocating Twitter users. In *CoNLL*.

[18] J. Eisenstein, B. O'Connor, N. Smith, and E. Xing. 2010. A latent variable model for geographic lexical variation. In *EMNLP*.

[19] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo. 2021. SATAR: A self-supervised approach to Twitter account representation learning and its application in bot detection. In *CIKM*.

[20] X. Feng, Y. Liang, X. Shi, D. Xu, X. Wang, and R. Guan. 2017. Overfitting reduction of text classification based on AdaBELM. *Entropy* 19, 7 (2017), 330.

[21] C. Fink, J. Kopecky, and M. Morawski. 2012. Inferring gender from the content of tweets: A region specific example. In *ICWSM*.

[22] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).

[23] B. Han, P. Cook, and T. Baldwin. 2012. Geolocation prediction in social media data by finding location indicative words. In *COLING*.

[24] B. Han, P. Cook, and T. Baldwin. 2013. A stacking-based approach to Twitter user geolocation prediction. In *ACL: System Demonstrations*.

[25] B. Han, A. Rahimi, L. Derczynski, and T. Baldwin. 2016. Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *WNUT*.

[26] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda. 2018. Back-translation-style data augmentation for end-to-end ASR. In *SLT*.

[27] B. Hecht, L. Hong, B. Suh, and E. Chi. 2011. Tweets from Justin Bieber's heart: The dynamics of the location field in user profiles. In *SIGCHI*.

[28] J. Hinds and A. Joinson. 2018. What demographic attributes do our digital footprints reveal? A systematic review. *PloS One* 13, 11 (2018), e0207112.

[29] B. Huang and K. Carley. 2017. On predicting geolocation of tweets using convolutional neural networks. In *SBP-BRiMS*.
[30] B. Huang and K. Carley. 2019. A hierarchical location prediction neural network for Twitter user geolocation. *arXiv preprint arXiv:1910.12941* (2019).
[31] D. Jurgens. 2013. That's what friends are for: Inferring location in online social media platforms based on social relationships. In *ICWSM*.
[32] A. Karimi, L. Rossi, and A. Prati. 2021. AEDA: An easier data augmentation technique for text classification. *arXiv preprint arXiv:2108.13230* (2021).
[33] D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[34] R. Krishnamurthy, P. Kapanipathi, A. Sheth, and K. Thirunarayan. 2015. Knowledge enabled approach to predict the location of Twitter users. In *ESWC*.
[35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[36] Y. Liu and L. Singh. 2021. Age inference using a hierarchical attention neural network. In *CIKM*.
[37] Yaguang Liu and Lisa Singh. 2023. Combining vs. transferring knowledge: Investigating strategies for improving demographic inference in low resource settings. In *WSDM*. 868–876.
[38] Y. Liu, L. Singh, and Z. Mneimneh. 2021. A comparative analysis of classic and deep learning models for inferring gender and age of Twitter users. In *DeLTA*.
[39] J. Mahmud, J. Nichols, and C. Drews. 2012. Where is this tweet from? Inferring home locations of Twitter users. In *ICWSM*.
[40] A. Mikołajczyk and M. Grochowski. 2018. Data augmentation for improving deep learning in image classification problem. In *IIPhDW*.
[41] Y. Miura, M. Taniguchi, T. Taniguchi, and T. Ohkuma. 2016. A simple scalable neural networks based model for geolocation prediction in Twitter. In *WNUT*.
[42] Y. Miura, M. Taniguchi, T. Taniguchi, and T. Ohkuma. 2017. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In *ACL*.
[43] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. 2016. How transferable are neural networks in NLP applications? *arXiv preprint arXiv:1603.06111* (2016).
[44] D. Nguyen, R. Gravel, D. Trieschnigg, and T. Meder. 2013. "How old do you think I am?" A study of language and age in Twitter. In *ICWSM*.
[45] J. Pennington, R. Socher, and C. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*.
[46] L. Perez and J. Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).
[47] A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021).
[48] A. Rahimi, T. Cohn, and T. Baldwin. 2018. Semi-supervised user geolocation via graph convolutional networks. *arXiv preprint arXiv:1804.08049* (2018).
[49] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta. 2010. Classifying latent user attributes in Twitter. In *NIPS MLSN Workshop*.
[50] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldridge. 2012. Supervised text-based geolocation using language models on an adaptive grid. In *EMNLP*.
[51] K. Ryoo and S. Moon. 2014. Inferring Twitter user locations with 10 km accuracy. In *WWW*.
[52] S. Sakaki, Y. Miura, X. Ma, K. Hattori, and T. Ohkuma. 2014. Twitter user gender inference using combined analysis of text and image processing. In *VL*.
[53] T. Sakaki, M. Okazaki, and Y. Matsuo. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *WWW*.
[54] L. Singh, M. Traugott, L. Bode, C. Budak, P. Davis-Kean, R. Guha, J. Ladd, Z. Mneimneh, Q. Nguyen, J. Pasek, T. Raghunathan, R. Ryan, S. Soroka, and L. Wahedi. 2020. Data blending: Haven't we been doing this for years? *Georgetown Massive Data Institute Report* (2020). https://live-guwordpressmccourt.pantheonsite.io/wpcontent/uploads/2020/05/MDI-Data-Blending-White-Paper-April2020.pdf
[55] L. Sloan, J. Morgan, P. Burnap, and M. Williams. 2015. Who tweets? Deriving the demographic characteristics of age, occupation and social class from Twitter user meta-data. *PloS One* 10, 3 (2015), e0115545.
[56] Twitter. 2023. *Twitter Decahose*. Retrieved from https://developer.twitter.com/en/docs/twitter-api/enterprise/decahose-api/overview/decahose
[57] P. Vijayaraghavan, S. Vosoughi, and D. Roy. 2017. Twitter demographic classification using deep multi-modal multi-task learning. In *ACL*.

[58] J. Wei and K. Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* (2019).
[59] Wikipedia. 2023. Houston, Texas. Retrieved from https://en.wikipedia.org/wiki/Houston
[60] B. Wing and J. Baldridge. 2014. Hierarchical discriminative classification for text-based geolocation. In *EMNLP*.
[61] Z. Wood-Doughty, N. Andrews, R. Marvin, and M. Dredze. 2018. Predicting Twitter user demographics from names alone. In *Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*.
[62] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL*.
[63] X. Zheng, J. Han, and A. Sun. 2018. A survey of location prediction on Twitter. *IEEE Trans. Knowl. Data Eng.* 30, 9 (2018), 1652–1671.

Received 16 May 2023; revised 9 April 2024; accepted 31 May 2024