GEO-SEQ2SEQ: Twitter User Geolocation on Noisy Data through Sequence to Sequence Learning

Jingyu Zhang♠ **Alexandra DeLucia**♠ **Chenyu Zhang**♣ **Mark Dredze**♠

♠Department of Computer Science, Johns Hopkins University
{jzhan237, aadelucia, mdredze}@jhu.edu
♣Department of Computer Science, Stanford University
chenyuz@stanford.edu

Abstract

Location information can support social media analyses by providing geographic context. Some of the most accurate and popular Twitter geolocation systems rely on rule-based methods that examine the user-provided profile location, which fail to handle informal or noisy location names. We propose GEO-SEQ2SEQ, a sequence-to-sequence (seq2seq) model for Twitter user geolocation that rewrites noisy, multilingual user-provided location strings into structured English location names. We train our system on tens of millions of multilingual location string and geotagged-tweet pairs. Compared to leading methods, our model vastly increases coverage (i.e., the number of users we can geolocate) while achieving comparable or superior accuracy. Our error analysis reveals that constrained decoding helps the model produce valid locations according to a location database. Finally, we measure biases across language, country of origin, and time to evaluate fairness, and find that while our model can generalize well to unseen temporal data, performance does vary by language and country.

https://github.com/JHU-CLSP/
Geo-Seq2seq-Twitter

1 Introduction

The analysis of Twitter and other social media data supports research in numerous domains by providing a measure of population beliefs and behaviors. A key aspect of many of these studies is the contextualization of posts based on the users' location. For example, studies of COVID-19 social distancing rely on knowing the location of users and how they move over time (Xu et al., 2020), models of disease spread during pandemics utilize updated information on population movements (Dredze et al., 2016), and studies of civil unrest and democratic reforms rely on isolating data from specific geographic areas (Sech et al., 2020; Chinta et al., 2021; Alsaedi et al., 2017; Littman, 2018). However,
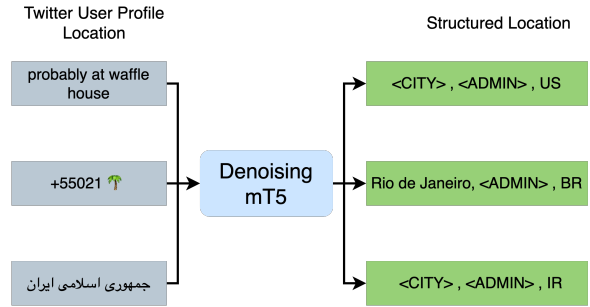

Figure 1: Geolocation of a user profile location to structured location. For example, GEO-SEQ2SEQ correctly maps "waffle house" (a US-based restaurant) to the US, a zip code to Brazil, and the Farsi name for Iran to Iran.

while some data contains user-provided structured location information, most do not. Furthermore, it is increasingly difficult to rely exclusively on the available location tweet metadata. Location-specific information has slowly been losing popularity among users, and Twitter has followed suit by removing the ability to add precise coordinates altogether (Kruspe et al., 2021). Previous metadata analyses studies have validated this trend and found a decline in user-provided location information, i.e., coordinates (stopped in 2019) and place objects (declining and only available in 2% of tweets) (Zhang et al., 2022; Kruspe et al., 2021).

Therefore, many researchers rely on Twitter geolocation systems, which automatically infer the location of a user or a tweet. Most approaches to social media user geolocation utilize tweet- or user-level metadata (Dredze et al., 2013), tweet content (including hashtags) (Alsaedi et al., 2017; Rahimi et al., 2016; Han et al., 2014; Wu and Gerber, 2018), and social networks (Rout et al., 2013; Jurgens, 2013). These systems examine one or more tweets from one user and resolve the tweet or user to a structured location object from a gazetteer, or a geographical dictionary, such as Google Maps

or GeoNames.[1] Researchers can then filter to a location of interest, or contextualize information based on locations.

A drawback of this method is the reliance on hand-crafted rules, which do not cover the diverse range of ways in which users specify locations. Not all users fill out the location field for its intended use or may put inaccurate locations (Hecht et al., 2011), slang location names, or a variety of other location strings that may be identifiable to people but not to rule-based string matching systems.

Rather than rely on existing string matching approaches, we propose to *learn* a sequence-to-sequence (seq2seq) model that maps noisy, multilingual, user-provided location strings into a structured location object selected from a database. For example, our system learns that `Windy City` corresponds to the location object `Chicago, IL, US` and `Zhongguo` refers to `China` (see Figure 1 for more examples). We train our system on tens of millions of tweets that contain both user-authored location profile strings and user-provided structured location information. We integrate our seq2seq model with Carmen (Dredze et al., 2013), a popular Twitter geolocation tool to produce a unique location in the GeoNames location database.[2] We build on mT5 (Xue et al., 2021) and experiment with various types of restrictions (constraints) in a denoising Transformer-based seq2seq model, including a trie-based constrained decoding (De Cao et al., 2021) scheme to ensure the output corresponds to a known location. We find that our system achieves better accuracy and greatly expanded coverage compared to existing systems. Finally, inspired by Zhang et al. (2022), we evaluate the fairness of our model with respect to performance across languages, country of origin, and time.

We make the following contributions:

- GEO-SEQ2SEQ, A denoising Twitter user geolocation model that learns to map user profile location strings to locations in a database.

- TWITTER-PUG, A dataset of multilingual, noisy strings paired with their location output mined from 35.4M Twitter user profile location string – true location pairs.

- An analysis of model biases in performance across language, country, and time.

2 Twitter Geolocation

Twitter geolocation tools can focus either on *user* geolocation (i.e., where is this user based) or *tweet* geolocation (i.e., where was this tweet written). Additionally, a system can examine a single tweet or all information about a user. Our focus is on ascertaining a user's primary location based on their profile information, which remains constant across all of their tweets but can be extracted from a single tweet.

The location of a *tweet* can be identified through: (1) the coordinates embedded in the metadata of the tweet (Dredze et al., 2013; Zhang et al., 2022), (2) the user-provided place metadata (Dredze et al., 2013; Zhang et al., 2022), and (3) inference from the tweet content (Alsaedi et al., 2017; Rahimi et al., 2016; Han et al., 2014; Wu and Gerber, 2018; Halterman, 2017; Izbicki et al., 2019).

The location of a *user* can be ascertained through (1) aggregated tweet locations from many geotagged or location-inferred tweets (from previously mentioned methods), (2) the user's location string in their profile (Dredze et al., 2013; Zhang et al., 2022), or (3) social network information (Rout et al., 2013; Jurgens, 2013).

The Carmen (Dredze et al., 2013; Zhang et al., 2022) geolocation tool infers a location for a user from a single tweet by looking for place metadata, provided coordinates, and (mostly) using a rule-based parser that maps a profile user location string to an internal location database based on GeoNames. We will utilize this rule-based parser as a comparative baseline for our method.[3]

Since our focus is on using the user profile location string alone, we omit comparisons to geolocation systems that use other methods. We choose this approach due to speed, privacy, and the prominence of location profile data. GEO-SEQ2SEQ is fast because the input is only the user profile location string, as opposed to requiring multiple tweets from a user for content analysis or making numerous Twitter API calls to gather a user's friends for a social network analysis. Further, this method can work on any pre-collected tweets with user profile information, which is advantageous due to the March 2023 depreciation of the free API tier.[4] Regarding privacy, we only use information

---

[1] https://www.geonames.org/

[2] We use Carmen 2.0, which provides greater location coverage by using the GeoNames database (Zhang et al., 2022).

[3] While Carmen is rule-based, its location aliases are learned from a network analysis.

[4] https://twitter.com/TwitterDev/status/1641222782594990080

freely provided by the user. We discuss this further in Section 9. Finally, as shown by Kruspe et al. (2021); Zhang et al. (2022), profile location strings are the only location-related metadata consistently provided by users through the years (60%), unlike Place and precise coordinates, which are provided 2% of the time and have been removed, respectively.

For these reasons, the comparison methods outlined above are not relevant. Further, while other methods have provided baselines against TWITTER-WORLD and TWITTER-US (Han et al., 2012), these datasets are English-only, so including them in this work would not demonstrate the multilingual ability of our approach.

3 Data

The goal of our system is to map a free text string from a user's profile location field to a known structured place name. The location field contains diverse types of content (see Figure 1), some of which may map to a specific city, or only a country, or no known place. We will learn these mappings based on a large corpus of historical Twitter data.

Common practice is to treat the **Place** object in the tweet metadata as the ground truth location of a user (see Figure 2).[5] This metadata is included when a user chooses to add it to their tweet. It contains a formal place name of a city, an administrative region (e,g. state), or a country.[6] However, only 2% of the tweets contain a Place object (Kruspe et al., 2021; Zhang et al., 2022). The Place object is an accurate but scarce source of geolocation information. In comparison, 30% to 40% of users (amounting to 60% of tweets) fill out the Twitter **profile** location string. Being a free-text field completed by the user, the profile string contains informal location names, made-up locations, or jokes by the users. The profile string is a noisy but abundant geolocation information source.

We frame our task as a *supervised learning problem where the goal is to translate the noisy profile location string into the structured place object* (or an equivalent representation). By collecting tweets

with both, we can create a large supervised dataset for training.

3.1 Geolocation Dataset

We create the Twitter Paired User Geolocation (TWITTER-PUG) dataset composed of 35.4M pairs of user profile location strings and formatted Place objects. We built this dataset by using the Twitter API to collect geotagged tweets worldwide. The tweets come from three different drawn bounding boxes, designed to cover the entire world, similar to the TWITTER-GLOBAL dataset from Zhang et al. (2022). The tweets are from 2013 to 2021. However, in these special geotagged streams, only tweets with a Place object or coordinates are included, as opposed to all tweets in the random stream. We select tweets that (1) contain a Twitter Place object in the metadata (some older geotagged tweets contain coordinates only) and (2) are posted by a user with a non-empty user profile location. While our model runs inference on just the location string, geotagged tweets with place metadata are needed for supervised training as the ground truth labels. To eliminate potential duplicates and bias introduced by prolific tweeters, we filtered the dataset to only one tweet per user. Since users can tweet from multiple locations (e.g., while traveling), which introduces noisy labels, we use the most common tweet location as the ground truth.

We represent the ground truth as a formatted string built directly from the tweet's Place object. The Place object contains information about the city, the administrative region, and the country of the tweet. In order for the model to learn the expected formatting of place names, we include special tokens `<CITY>`, `<ADMIN>`, `<COUNTRY>` in any missing fields. For ease of use in the multilingual dataset, we only include the ISO 3166-1 alpha-2 country codes instead of the full country name, such as "US" for "United States." An example of the derived location string is in Figure 2.

The final dataset contains 35.4M profile string and structured ground-truth string pairs. We sampled 33.4M for the training set, and 1M for validation and test, respectively. We provide more details in Appendix B, with language and country distribution info in Figure 6.

2022 Dataset Since geotagging behavior may change over time (Zhang et al., 2022) and exhibits biases (Pavalanathan and Eisenstein, 2015), we evaluate our model on an additional collection of

---

[5]Pavalanathan and Eisenstein (2015) highlight biases in relying on users who provide location data as being representative of all users, and Wood-Doughty et al. (2017) show that different types of people use the platform differently. We include several measures of geolocation fairness to explore some of these effects.

[6]A small number of places can contain neighborhoods or points of interest.
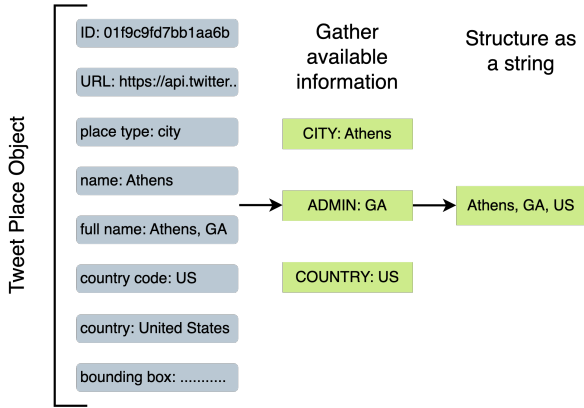
Figure 2: Ground truth label created from the tweet place objects. Each ground truth string is of the form "`<CITY>,<ADMIN>,<COUNTRY>`." The special tokens are left as-is when information is not available or does not apply.

unseen users from the 2022 public stream as an out-of-distribution test set. To preserve the distribution of the public stream, we do not conduct user deduplication on this test set. The 2022 evaluation dataset contains 588K geotagged tweets.

3.2 Location Database

Twitter user geolocation maps a user location string to an entry in a location database. We use the **GeoNames-combined** database from Zhang et al. (2022), which combines location entries derived from Twitter places with entries in the GeoNames gazetteer with populations over 15K, and contains a total of 73,921 entries.

4 Methods

We utilize an encoder-decoder transformer-based model to learn a mapping from user location profile strings to structured place strings. Given the multilingual nature of our dataset, we select the multilingual T5 model (mT5) (Xue et al., 2021). As discussed in Section 3, we add three special tokens: `<CITY>`, `<ADMIN>`, `<COUNTRY>`, and fine-tune the embeddings[7] for these tokens along with the model. We fine-tune mT5-small for our task on the 33.4M training examples with the Adam optimizer for cross-entropy loss for 5 epochs. All decoding methods use the same pretrained model unless stated otherwise. Training details are in Appendix A. We call our model GEO-SEQ2SEQ.

---

[7]Initialized with the default Hugging Face settings of random weights.

4.1 Trie-Based Constrained Decoding

A trained GEO-SEQ2SEQ model computes the conditional probability $p(y \mid s)$ of a formal location name y given a user profile string s. To produce the best candidate location y^*, ideally, we would enumerate every location name defined in the location database $y \in \mathcal{D}$, and choose the best scoring one $y^* = \arg\max_{y \in \mathcal{D}} p(y \mid s)$. However, this is intractable due to the size of our location database. Instead, we turn to beam search (Sutskever et al., 2014) to approximate the best-scoring candidate in a tractable manner.

Because we assume a finite set of possible locations as defined by our location database, we incorporate this prior knowledge in the inference stage of GEO-SEQ2SEQ by forcing the seq2seq model to generate a valid location. We employ constrained beam search (De Cao et al., 2021) where the constraint is in the form of a trie (i.e., a prefix-tree).[8] The tree-like structure in a trie is a natural fit to efficiently organize a large set of location names because they are inherently hierarchical. We build the trie using the set of all location names in the database. An example of the trie is shown in Figure 3. The trie is divided into different country-level sub-tries (e.g., sub-tries rooted by tokens `US`, `CA`), and each country sub-trie contains admin-level sub-tries (e.g., the `US-Colorado` and `US-Montana` sub-tries).

To perform trie-based constrained beam search, at each decoding timestep, the current state corresponds to a node $t \in \mathcal{T}$ on the trie (starting from `<BOS>`$\in \mathcal{T}$ as the first token). To select the next candidate token, only the tokens that are children of t are allowed. A beam is considered complete when the current state has no children (when the `<EOS>` token is reached).

In related work, constrained decoding has also been utilized in other tasks with structured output, such as entity retrieval (De Cao et al., 2021), event extraction (Lu et al., 2021), parallel sentence mining (Chen et al., 2020), and dependency parsing (Li et al., 2018). Ou et al. (2021) use a disjunctive lexical constraint to guide generation within frame semantics (Fillmore, 1976). Mao et al. (2020) use constrained decoding to preserve factual consistency in abstractive summarization. To the best of our knowledge, GEO-SEQ2SEQ is the first method

---

[8]We use the Matching Algorithm with Recursively Implemented StorAge (MARISA) data structure for trie implementation. http://www.s-yata.jp/marisa-trie/docs/readme.en.html
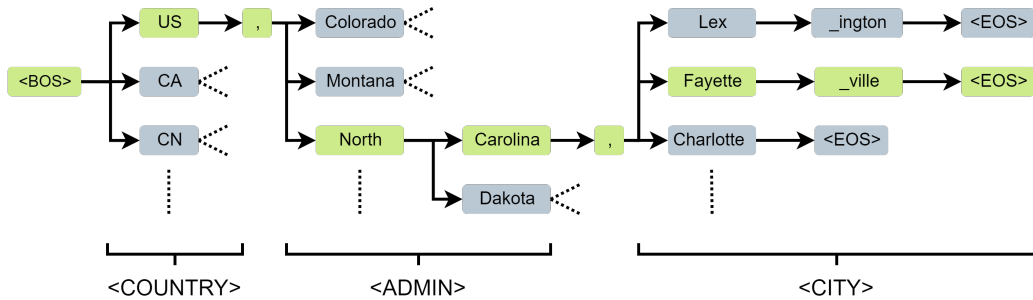
Figure 3: Excerpt from the "reversed" decoding trie built from the Carmen location database. The output sequence is constrained at each overarching step to <BOS> → <COUNTRY> → ... <EOS>. At each sub-step, the generated tokens are constrained to valid subwords, or those present in the location database at that step.

that applies constrained decoding techniques on the task of Twitter user geolocation.

4.2 Reversing the Output

We format the Place object as the string <CITY>, <ADMIN>, <COUNTRY>. However, from a decoding standpoint, this is backwards. Intuitively, we can most easily guess a country for a tweet, then select an admin conditioned on the country, and a city conditioned on the admin and country. It may be beneficial to instead generate the reverse of our Place string so it is from higher to lower granularity: <COUNTRY>, <ADMIN>, <CITY>. The trie in Figure 3 is reversed. The reverse trick has two advantages: (1) the resulting constraint trie is more compact since the hierarchical order of location names is followed and (2) the seq2seq model is not required to generate the correct city at the beginning of decoding, which is difficult. The decoding of <ADMIN> can attend to the generated <COUNTRY> slot, and the decoding of <CITY> can attend to country and admin-level information. We apply this reverse trick in tandem with the constrained decoding methods.

5 Comparison Methods

We include several baseline methods for comparison against our proposed model.

Table Lookup Baseline The power of a seq2seq model is in its ability to not just memorize input-output strings, but to infer output from previously unseen input sequences. We directly test our model against this simple memorization baseline. Using the training data, we built a dictionary mapping user profile locations to the formatted output string, and the "prediction" from this baseline is a dictionary lookup. If an input has more than one output (which occurs for 20% of training data), then the

output is uniformly sampled from the associated possible outputs. If the input is not found in the dictionary, then the prediction is treated as null and counted against the model's performance.

Carmen Profile Resolver As discussed in Section 2, Carmen has a simple rule-based method to match an input profile location string to a known location in its internal database. Specifically, the Carmen profile resolver normalizes user location strings through rules such as stripping punctuation and collapsing runs of whitespace, and matches the normalized string with location names in the database.

Carmen + GEO-SEQ2SEQ Carmen accurately matches many simple location strings to the correct location, but fails to handle more complex strings. In contrast, GEO-SEQ2SEQ can handle any string. We evaluate a hybrid approach in which we first use Carmen's rule-based strategies (profile resolver) and apply GEO-SEQ2SEQ to strings that were not resolved by Carmen. This approach is the preferred use case, as rule-based methods are faster than inferencing with mT5, even with the small model.

Ground Truth We feed the ground truth structured output sequence (target) directly to Carmen, which measures the ability of the resolver to match the official location name to an entry in the locations database. This is considered an approximate upper bound of denoising model performance; the best we could hope from our model is to perfectly reconstruct the official place name. We do not achieve perfect accuracy for the ground truth, especially on the city level, due to several reasons: (1) The location database does not contain every location on earth. The database was constructed to include all cities with at least 15k inhabitants. (2) The name of a location is not unique. Some

locations have multiple names due to historical or political reasons. (3) The ground truth location names are in various languages, and although the location database contains alternative location names in many languages, this set of aliases is not exhaustive.

6 Evaluation

We evaluate all models from three perspectives: coverage, geolocation accuracy, and the validity rate of generated location strings.

Coverage We define *coverage* as the fraction of tweets that were resolved to a location. A tweet is "resolved" if the geolocation system successfully proposed a candidate location given the user location string. The *coverage* metric is similar to recall, but does not consider whether the prediction is correct.

Geolocation Accuracy To evaluate the correctness of resolved tweets, we use the *accuracy* metrics from Zhang et al. (2022). Specifically, we use the *match ratio* metric (denoted mr) to evaluate whether the candidate location matches the ground truth on the city, admin, or country level. We make one change to ensure a fair comparison: instead of calculating the match ratio over the resolved locations, which are different sets of locations for different candidate systems, we calculate over all test tweets, which ensures the same denominator across all matching ratio scores. We also ensure that a model is not penalized for not guessing a city or admin when no city/admin was provided by awarding credit for the <CITY> and <ADMIN> tokens.

Validity Rate Hallucination is a known challenge for neural text generation models (Dziri et al., 2022; Ji et al., 2022). Since our GEO-SEQ2SEQ approach is at risk of hallucination, we evaluate the *validity rate* of the generated location names on the country, admin, and city levels. The validity rate (denoted vr) is the fraction of test examples where the generated string is a valid location name (i.e., it matches with one of the location names in the location database). Measuring validity is more important for the non-constrained methods (non-trie), as it is not possible for the model to generate an invalid location with the trie (see Section 4.1).

7 Experimental Results

We evaluate the generalization effectiveness of the best version of GEO-SEQ2SEQ (constrained decoding with beam size of 16; see ablation results in Section 7.1) by comparing it to other methods on our geolocation dataset. GEO-SEQ2SEQ greatly outperforms the rule- and memorization-based models, showing that our model has learned to generalize to unseen locations.

With respect to coverage, the rule-based Carmen profile resolver performs the worst, followed by the Table Lookup baseline, only providing locations for 53% and 82%, respectively (see results in Table 2). Surprisingly, the Carmen-integrated model and GEO-SEQ2SEQ slightly outperform the Ground Truth upper bound on performance, indicating that the model learned patterns from other strings that are more useful than the original Twitter place names (i.e., ground truth).

The remaining metrics evaluate accuracy with respect to geolocation and structured prediction format, specifically whether the output is in the correct <CITY>, <ADMIN>, <COUNTRY> form, and whether each slot contains a location in the Carmen location database. Note that the output from GEO-SEQ2SEQ and the Carmen-augmented model achieve a perfect score of 1.0 because they are forced to output valid locations through constrained decoding. The non-constrained methods, Table Lookup and Ground Truth, have similar validity rates due to being based on Twitter Place names, not all of which are present in Carmen's location database.

With regards to geolocation accuracy, we look at the *match ratio* for each country, admin, and city slot. While GEO-SEQ2SEQ by itself has a very high accuracy of 85% and 68% for country and admin, respectively, the Carmen-augmented model has higher city accuracy at 34% (versus 31%). This improvement in granularity at the city level suggests the integrated model is better suited for tasks that require finer demographic granularity.

7.1 Ablation Study

Our main results shows GEO-SEQ2SEQ with constrained decoding with beam size 16. To determine which components of our model were most effective, we run an ablation study over different decoding methods (greedy, beam search, trie-based constrained beam search), whether the reverse trick is utilized, and whether to use Carmen along with

Method	Coverage	$mr_{country}$	mr_{admin}	mr_{city}	$vr_{country}$	vr_{admin}	vr_{city}	vr_{format}
Ground Truth	.934	.934	.919	.117	.995	.774	.659	.995
Table Lookup Baseline	.836	.499	.207	.002	.847	.706	.668	1.000
Carmen profile	.494	.452	.290	.152	-	-	-	-
GEO-SEQ2SEQ (Best)	.994	.778	**.689**	.256	1.000	1.000	1.000	1.000
Carmen + GEO-SEQ2SEQ	**.995**	**.831**	.593	**.437**	1.000	1.000	1.000	1.000

Table 1: Results for GEO-SEQ2SEQ in comparison to other methods on newer Tweets from 2022.

Method	Coverage	$mr_{country}$	mr_{admin}	mr_{city}	$vr_{country}$	vr_{admin}	vr_{city}	vr_{format}
Ground Truth	.984	.984	.961	.405	.998	.781	.577	1.000
Table Lookup Baseline	.816	.594	.229	.001	.828	.722	.650	1.000
Carmen Profile Resolver	.527	.468	.245	.135	-	-	-	-
GEO-SEQ2SEQ (Best)	.992	**.845**	**.679**	.309	1.000	1.000	1.000	1.000
Carmen + GEO-SEQ2SEQ	**.994**	.840	.540	**.342**	1.000	1.000	1.000	1.000

Table 2: Results for GEO-SEQ2SEQ in comparison to other methods. Carmen + GEO-SEQ2SEQ is how an enhanced Carmen would be used in practice.

GEO-SEQ2SEQ. Results appear in Table 3. We notice that the coverage is consistently high ($>.99$) over all ablation settings. Therefore, we discuss the match ratio and validity rate of different settings below.

Decoding Method In addition to trie-based constrained beam search, we experiment with greedy decoding and unconstrained beam search with beam size 16.[9] In terms of the accuracy metrics, we find the match ratio for greedy and beam search are largely similar. Interestingly, the trie-based constrained decoding setting greatly outperforms greedy and beam search in mr_{city}. We hypothesize this is because for constrained decoding, once the country and admin are generated correctly, it is relatively easy to select the correct city from a small set of city names within a particular administrative region, in comparison to the unconstrained scenario where the model can generate any string. However, unconstrained beam search slightly outperforms the constrained decoding setting on mr_{admin}. In terms of validity rates, while beam search outperforms greedy decoding on vr_{admin}, greedy decoding is slightly superior on vr_{city}.

The Reverse Trick The forward and reverse variants of GEO-SEQ2SEQ have largely comparable performance. While the reverse variants perform slightly better on the match ratio metrics (with the exception of mr_{admin}), the forward variants have slightly higher validity rates.

[9]In a preliminary experiment, we varied the beam size between 8, 16, and 32, and found little difference between the results.

Combining with Carmen We see a comparable $mr_{country}$, slightly worse mr_{admin}, and notably better mr_{city}. On validity rates, the combination achieves higher vr_{admin} but lower vr_{city}.

7.2 Qualitative Examples

Figure 4 shows examples of GEO-SEQ2SEQ on the test set, displaying the input string, ground truth (reversed), and the model's output. A qualitative review finds four categories of instances: "ideal" match, non-English, mismatched, and fictional/joke.

Ideal locations are unambiguous from the profile string, and can easily be matched with high accuracy. While Boca Raton, Florida, US is a perfect match, we see that "California" is matched to San Diego as opposed to it's ground truth of Anaheim. This is understandable, as no information beyond the state (admin) was provided, and the model is correct on the country and admin levels. The second category is composed of location strings that match their ground truth location, but are in a language other than English. In this situation, the multilingual pretraining of mT5 is very helpful.

The last two categories are predominantly noisy, as they consist of mismatched location string and ground truth pairs, or completely fictional or joke locations. Mismatched string–place pairs often result from users on vacation, or users who are away from their home for many reasons. Fictional locations are those that do not exist and are either jokes or references to popular culture (e.g., "bikini bottom" from SpongeBob SquarePants and "221B Baker Street" from Sherlock Holmes). Since GEO-

Method	Coverage	$mr_{country}$	mr_{admin}	mr_{city}	$vr_{country}$	vr_{admin}	vr_{city}	vr_{format}
Forward Greedy	.987	.848	.744	.251	1.000	.799	.775	1.000
+Carmen	.992	.842	.557	.333	.999	.887	.709	1.000
Forward Beam	.985	.851	**.735**	.228	.999	.807	.718	1.000
+Carmen	.991	.844	.553	.332	.999	.906	.670	1.000
Forward Trie	.992	.845	.702	.319	1.000	1.000	1.000	1.000
+Carmen	.994	.839	.542	.335	1.000	1.000	1.000	1.000
Reverse Greedy	.985	**.866**	.600	.237	.999	.659	.680	1.000
+Carmen	.991	.855	.492	.340	.999	.732	.594	1.000
Reverse Beam	.985	.851	.725	.232	.999	.784	.673	1.000
+Carmen	.991	.845	.553	.340	.999	.889	.617	1.000
Reverse Trie	.992	.845	.679	.309	1.000	1.000	1.000	1.000
+Carmen	**.994**	.840	.540	**.342**	1.000	1.000	1.000	1.000

Table 3: Ablation experiment of the seq2seq resolver over decoding method and the reverse trick.

SEQ2SEQ always outputs a prediction, it usually is wrong about fictional/joke places. We discuss this further in Section 9.

7.3 Results "In the Wild"

While GEO-SEQ2SEQ was trained on a significant amount of data from 2013–2021, we wanted to ensure it could generalize to new temporal data. We test our method on the 2022 Dataset collected from the Twitter 1% stream (see Section 3). Despite the temporal shift, we see very similar performance when comparing GEO-SEQ2SEQ and Carmen+GEO-SEQ2SEQ on the test set (Table 2) to the new 2022 data (Table 1). Coverage remains at 99% for both models, and the trend of Carmen+GEO-SEQ2SEQ having better finer-granularity performance than GEO-SEQ2SEQ alone still holds.

8 Performance across Demographics

Metrics over the entire test set can hide biases in model behavior on specific sub-groups. When used as part of an analysis pipeline these biases could change study conclusions. For example, Han et al. (2012) exclude non-English tweets since location based on language ID (e.g. Japanese tweets come from Japan) may portray an unrealistic picture of model performance. We conduct a language and location analysis to determine the fairness of the best performing GEO-SEQ2SEQ model as measured on the test set.

Language Bias Does the language of the profile location bias model behavior? We define a prediction as "language-biased" if the predicted country's primary language, as identified by GeoNames, is the same as the language of the source location string. Since English is prevalent around the world, we remove countries that have English as a primary language for this experiment, leaving 115 out of
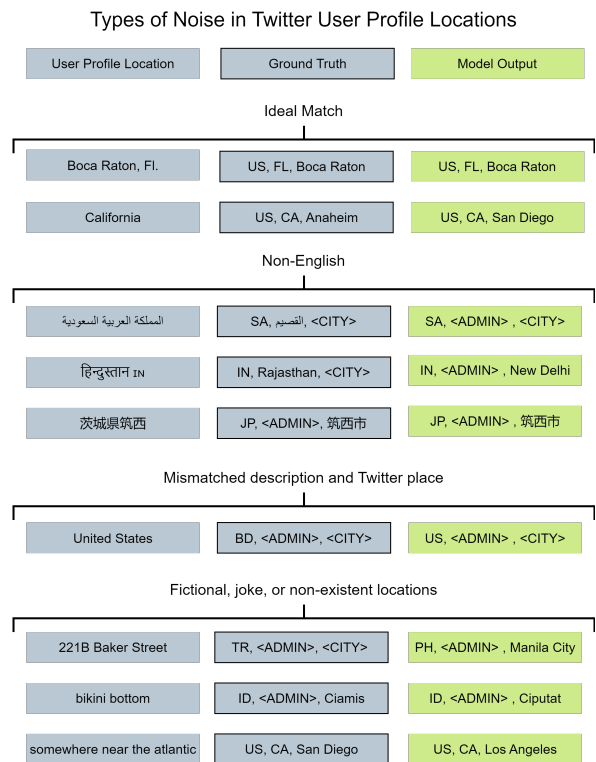

Figure 4: Qualitative examples in the TWITTER-PUG test set predicted by the best GEO-SEQ2SEQ model. Input strings can be categorized into "ideal", non-English, mismatched, or fictional/joke categories.
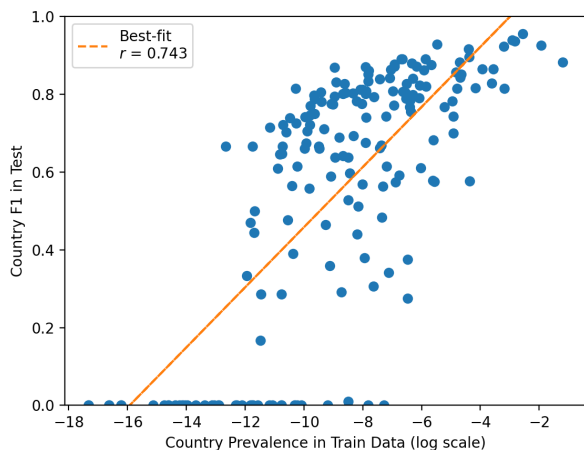
Figure 5: Prevalence (log of frequency) of examples for each country in the train data plotted against the F1 score for that country in the test set. Pearson's r of 0.743 shows a strong correlation between the amount of training data per country and the model's performance.

146 countries.[10] The list of countries included in the analysis is in Appendix Appendix D.

Our model predicted one of the remaining 115 countries for 538k test set examples, and we identified 244k as "language-biased." Among the language-biased predictions, 231k (94%) are "correctly biased," meaning the predicted country correctly matches the target country. Thus, only 6% of the predictions are wrongly biased by the language of the profile location. Further analysis is needed in languages prevalent in multiple countries.

Fairness in Performance   We next measure the fairness of predictions at the country level across languages and countries. For the per-country performance, we calculate F1 for each country by treating each country as its own "class" (Appendix Figure 7a). We removed countries with less than two examples in the test set (bottom 10th percentile), leaving 169 out of 185 countries. There is a large gap between countries with high F1 (around 94%: India, Turkey, Japan) and those with low F1 (0%). 25 countries had 0% F1, 15 of which are European countries (e.g., Netherlands, and Ireland). Most of these predictions are incorrectly mapped to the US. This discrepancy could be due to noise from mismatched location string and ground truth or low volume of those countries present in the training data (i.e., less than 0.01%).

We then analyze how much the availability of each country's training data affects the prediction

accuracy of GEO-SEQ2SEQ (Figure 5). The Pearson correlation coefficient between data availability and country F1 across all our countries is 0.743, indicating a strong correlation between the two and suggesting that a reason for the low F1 for some countries could be their insufficient presence in the training data.

For the per-language performance, we use a basic accuracy metric, based on $mr_{country}$, but aggregated by the language tag provided in the Twitter metadata.[11] As in the per-country analysis, we remove the bottom 10th percentile of languages, filtering from 69 to 62 languages. The score is broken down for each language in Figure 7b. Similar to the performance across countries, there is a large discrepancy in performance across languages with the highest (98%: Marathi, Gujarati, and Kannada) and lowest accuracy (65%: French, Lao, and Italian). Indic languages have the highest accuracy, perhaps because they are the most concentrated by location. The first non-Indic language with high accuracy is Turkish (93%) followed by Japanese (92%). We discuss possible strategies to better support all languages and countries in Limitations.

9   Conclusion

We present GEO-SEQ2SEQ, an mT5 model fine-tuned for Twitter user geolocation through denoising user profile location strings. We train it on TWITTER-PUG, a dataset of 35.4M location strings with ground truth labels. Our model outperforms existing systems with 99% test set coverage and 85% prediction accuracy at the country-level. Augmented with Carmen the model achieves 34% city accuracy, improving over Carmen's 14% accuracy. The success of the model comes from a constrained decoding strategy with a beam size of 16, with a "reversed" target string. Additionally, we breakdown performance by location and language, highlighting biases in model behavior. Future work should concentrate on producing models that are fairer with regard to locations and languages.

---

[10]Out of 185 countries present in the test set, only 146 were predicted by GEO-SEQ2SEQ.

[11]Since F1 metrics are label-specific, they are not appropriate for this extraneous language label, since no language is being predicted.

Limitations

Ground Truth and Data Cleaning Although we conduct basic cleaning by selecting the ground truth Place object that has appeared the most often for a given user, this is only a heuristic and does not guarantee that the selected ground truth matches the description in the user location string, which introduces noise in the TWITTER-PUG dataset. Future work is needed to develop more accurate methods that identify the ground truth from a set of geotagged user tweets. Also, the current ground truth format does not account for alternative names in geolocation. A future direction is training the seq2seq model to generate multiple formal location names from a single user location string. Alternative names in gazetteers such as GeoNames could be used as a source of this ground truth.

In Figure 4, we identified several types of noise in Twitter user profile locations. We did not conduct extensive data cleaning of fictional, joke, or non-existent locations. Though we attempted to filter these places automatically, we found little change in model performance. A more detailed study of the effects of data cleaning would be beneficial.

Model Size Due to resource constraints, we only experiment with the mT5-small model. In a small-scale preliminary study, we found mT5 outperforms ByT5 (Xue et al., 2022) on our task of geolocation name transduction. It would be interesting to also test how larger (e.g. mT5-large) or other types of pretrained language models (e.g. fully autoregressive models) performs on this task. Also, how much data is actually needed to train the model.

Coverage v.s. Accuracy Trade-Off Another limitation of the GEO-SEQ2SEQ approach is that the model always produces a candidate location even when the input only contains a fictional location or does not contain a location at all. A potential solution for this is thresholding the model based on a log-probability threshold, and only producing a candidate location when the probability of a beam is high enough. Such thresholding method could serve to trade off coverage and accuracy.

A related issue is the accuracy at each granularity (i.e., country, admin, and city). The model performs significantly better at lower granularity, specifically at the country level (see Table 2). This is important for end-users to acknowledge if this tool is used for higher-stakes analysis such as natural disaster relief, versus such as studying vaccine opinions in different parts of the world.

Performance Across Demographics Finally, as shown in Section 8, our model has a wide range of performance with respect to F1 across countries, and a smaller discrepancy of accuracy across language. The strong multilingual performance is most likely from the original mT5 pre-training. However, there is still room for improvement. To address the discrepancy in performance across countries, a strategy is to stratify the data by country, similar to how multilingual pre-trained encoders are trained with exponential sampling based on language balance (Xue et al., 2021).

Ethical Considerations

The main ethical consideration for a tool like GEO-SEQ2SEQ is *privacy*. We respect user privacy in the creation of GEO-SEQ2SEQ as well as in collecting the data to build TWITTER-PUG by only using immediately available data provided by users. As discussed in Section 3, the training data is built from user profile location strings paired with a user's most frequently tagged Twitter Place. Once trained, GEO-SEQ2SEQ only needs the user profile location to run inference.

Also, due to the structured nature of the output string and easy integration with Carmen, researchers can easily choose at which granularity to aggregate their data, whether the city, admin (state/province), or country level.

Further, the use case of our model is only meant to support researchers studying location-specific demographics. The content will be studied in aggregate, as according to Twitter policy.

Acknowledgements

References

Nasser Alsaedi, Pete Burnap, and Omer Rana. 2017. Can We Predict a Riot? Disruptive Event Detection Using Twitter. *ACM Transactions on Internet Technology*, 17(2):1–26.

Pinzhen Chen, Nikolay Bogoychev, Kenneth Heafield, and Faheem Kirefu. 2020. Parallel sentence mining by constrained decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1672–1678, Online. Association for Computational Linguistics.

Abhinav Chinta, Jingyu Zhang, Alexandra DeLucia, Mark Dredze, and Anna L. Buczak. 2021. Study of Manifestation of Civil Unrest on Twitter. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 396–409, Online. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Mark Dredze, Manuel García-Herranz, Alex Rutherford, and Gideon Mann. 2016. Twitter as a source of global mobility patterns for social good. In *2016 ICML Workshop on Data4Good: Machine Learning in Social Good Applications*.

Mark Dredze, Michael J. Paul, Shane Bergsma, and Hieu Tran. 2013. Carmen: A Twitter Geolocation System with Applications to Public Health. Association for the Advancement of Artificial Intelligence.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Charles J Fillmore. 1976. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*, volume 280, pages 20–32. New York.

Andrew Halterman. 2017. Mordecai: Full Text Geoparsing and Event Geocoding. *Journal of Open Source Software*, 2(9):91.

B. Han, P. Cook, and T. Baldwin. 2014. Text-Based Twitter User Geolocation Prediction. *Journal of Artificial Intelligence Research*, 49:451–500.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Geolocation Prediction in Social Media Data by Finding Location Indicative Words. In *Proceedings of COLING 2012*, pages 1045–1062, Mumbai, India. The COLING 2012 Organizing Committee.

Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. 2011. Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 237–246. Association for Computing Machinery, New York, NY, USA.

Mike Izbicki, Vagelis Papalexakis, and Vassilis Tsotras. 2019. Geolocating Tweets in any Language at any Location. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 89–98, New York, NY, USA. Association for Computing Machinery.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

David Jurgens. 2013. That's What Friends Are For: Inferring Location in Online Social Media Platforms Based on Social Relationships. *Proceedings of the International AAAI Conference on Web and Social Media*, 7(1):273–282. Number: 1.

Anna Kruspe, Matthias Häberle, Eike J. Hoffmann, Samyo Rode-Hasinger, Karam Abdulahhad, and Xiao Xiang Zhu. 2021. Changes in Twitter geolocations: Insights and suggestions for future usage. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 212–221, Online. Association for Computational Linguistics.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Justin Littman. 2018. Charlottesville Tweet Ids. Publisher: Harvard Dataverse type: dataset.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yuning Mao, Xiang Ren, Heng Ji, and Jiawei Han. 2020. Constrained abstractive summarization: Preserving factual consistency with constrained generation. *CoRR*, abs/2010.12723.

Jiefu Ou, Nathaniel Weir, Anton Belyy, Felix Yu, and Benjamin Van Durme. 2021. InFillmore: Frame-guided language generation with bidirectional context. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*,

pages 129–142, Online. Association for Computational Linguistics.

Umashanthi Pavalanathan and Jacob Eisenstein. 2015. Confounds and consequences in geotagged Twitter data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2138–2148, Lisbon, Portugal. Association for Computational Linguistics.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2016. pigeo: A Python Geotagging Tool. In *Proceedings of ACL-2016 System Demonstrations*, pages 127–132, Berlin, Germany. Association for Computational Linguistics.

Dominic Rout, Kalina Bontcheva, Daniel Preoţiuc-Pietro, and Trevor Cohn. 2013. Where's @wally? a classification approach to geolocating users based on their social ties. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 11–20, New York, NY, USA. Association for Computing Machinery.

Justin Sech, Alexandra DeLucia, Anna L. Buczak, and Mark Dredze. 2020. Civil Unrest on Twitter (CUT): A Dataset of Tweets to Support Research on Civil Unrest. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 215–221, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zach Wood-Doughty, Michael Smith, David Broniatowski, and Mark Dredze. 2017. How does twitter user behavior vary across demographic groups? In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 83–89.

Congyu Wu and Matthew S. Gerber. 2018. Forecasting Civil Unrest Using Social Media and Protest Participation Theory. *IEEE Transactions on Computational Social Systems*, 5(1):82–94. Conference Name: IEEE Transactions on Computational Social Systems.

Paiheng Xu, Mark Dredze, and David A. Broniatowski. 2020. The Twitter Social Mobility Index:

Measuring Social Distancing Practices With Geolocated Tweets. *Journal of Medical Internet Research*, 22(12):e21499.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. ArXiv:2010.11934 [cs].

Jingyu Zhang, Alexandra DeLucia, and Mark Dredze. 2022. Changes in tweet geolocation over time: A study with carmen 2.0. In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022)*, pages 1–14, Gyeongju, Republic of Korea. Association for Computational Linguistics.

A Model Training and Inference Details

The GEO-SEQ2SEQ model is an mT5-small model fine-tuned on TWITTER-PUG for 5 epochs with cross-entropy loss. We use the Adam optimizer with a learning rate of 5e-5. The batch size for training is 96.

The training process took around 5 days to finish due to the massive amount of data in our collected TWITTER-PUG dataset. The decoding time on the main 1M test set of TWITTER-PUG varies for different decoding algorithms. While greedy decoding takes arouns 3 hours to decode, beam search with beam size 16 takes 13 and 6 hours for the trie-based constrained decoding and unconstrained decoding, respectively.

A single NVIDIA A100 GPU with 40GB memory is used for all experiments. We use the Hugging Face Transformers library for training and inference (Wolf et al., 2020).

B Dataset Details

In this section, we provide details of our collected TWITTER-PUG dataset. The detailed number of train, validation, and test examples are shown in Table 4. The language and country distribution plot is shown in Figure 6.

Due to the scale of data and the noisy nature of this task, we did not filter data for possible offensive content. While this is possible for English data, finding offensive-speech dictionaries in all 69 languages present in the data is difficult. However, a possible solution, mentioned in Section 9, is to ensure the model does not provide predictions for user profile location strings containing offensive content by restricting output if the log probability for the output is below a specific threshold.

A similar concern is of uniquely identifying information. While the user profile location string is meant to be filled in with a location, it can be completed with any string since it is free text. As in offensive speech detection and removal, identifying and removing possible names is difficult. However, since this data is collected from public profile information set by the user, uniquely identifying information is less of a concern.

C Additional Details on Performance across Demographics

Here we provide additional details on the performance across demographics. Figure 7a shows the

Split	Size
Train	33,416,150
Valid	1,000,000
Test	1,000,000
Total	35,416,150

Table 4: GEO-SEQ2SEQ dataset statistics.

F1 score with respect to country-level prediction for each country, and Figure 7b shows the country-level accuracy, $mr_{country}$, across languages.

D Performance across Demographics Details

For the **language bias** experiment, the following countries were identified as having English as a primary language: Australia (AU), Bahamas (BS), Botswana (BW), Canada (CA), Cameroon (CM), Fiji (FJ), Micronesia (FM), Ghana (GH), Gibraltar (GI), Gambia (GM), Guam (GU), Guyana (GY), India (IN), Kenya (KE), Liberia (LR), Lesotho (LS), Mauritius (MU), Namibia (NA), Nigeria (NG), New Zealand (NZ), Papua New Guinea (PG), Solomon Islands (SB), Seychelles (SC), Sierra Leone (SL), Eswatini (SZ), Uganda (UG), United States (US), British Virgin Islands (VG), U.S. Virgin Islands (VI), Zambia (ZM), and Zimbabwe (ZW). We used country information provided by GeoNames to identify these countries http://download.geonames.org/export/dump/countryInfo.txt. Countries with high numbers of second-language English speakers were also removed, (e.g., India). The remaining countries that were included in the analysis are shown in Table 5.

United Arab Emirates (AE)	Afghanistan (AF)	Armenia (AM)
Angola (AO)	Argentina (AR)	Azerbaijan (AZ)
Bangladesh (BD)	Burkina Faso (BF)	Bulgaria (BG)
Bahrain (BH)	Burundi (BI)	Benin (BJ)
Brunei (BN)	Bolivia (BO)	Bonaire, Saint Eustatius and Saba (BQ)
Brazil (BR)	Bhutan (BT)	Belarus (BY)
Democratic Republic of the Congo (CD)	Central African Republic (CF)	Republic of the Congo (CG)
Ivory Coast (CI)	Chile (CL)	China (CN)
Colombia (CO)	Cyprus (CY)	Djibouti (DJ)
Algeria (DZ)	Egypt (EG)	Spain (ES)
Ethiopia (ET)	France (FR)	Gabon (GA)
Georgia (GE)	French Guiana (GF)	Guinea (GN)
Equatorial Guinea (GQ)	Greece (GR)	Guinea-Bissau (GW)
Hong Kong (HK)	Indonesia (ID)	Israel (IL)
Iraq (IQ)	Iran (IR)	Italy (IT)
Jordan (JO)	Japan (JP)	Kyrgyzstan (KG)
Cambodia (KH)	Comoros (KM)	South Korea (KR)
Kuwait (KW)	Kazakhstan (KZ)	Laos (LA)
Lebanon (LB)	Sri Lanka (LK)	Latvia (LV)
Libya (LY)	Morocco (MA)	Madagascar (MG)
Marshall Islands (MH)	Mali (ML)	Myanmar (MM)
Mongolia (MN)	Macao (MO)	Northern Mariana Islands (MP)
Mauritania (MR)	Malta (MT)	Maldives (MV)
Malawi (MW)	Mexico (MX)	Malaysia (MY)
Mozambique (MZ)	New Caledonia (NC)	Niger (NE)
Netherlands (NL)	Norway (NO)	Nepal (NP)
Oman (OM)	Peru (PE)	Philippines (PH)
Pakistan (PK)	Portugal (PT)	Palau (PW)
Paraguay (PY)	Qatar (QA)	Reunion (RE)
Russia (RU)	Rwanda (RW)	Saudi Arabia (SA)
Sudan (SD)	Singapore (SG)	Senegal (SN)
Somalia (SO)	Suriname (SR)	Syria (SY)
Chad (TD)	Togo (TG)	Thailand (TH)
Tajikistan (TJ)	Timor Leste (TL)	Turkmenistan (TM)
Tunisia (TN)	Turkey (TR)	Taiwan (TW)
Tanzania (TZ)	Ukraine (UA)	Uruguay (UY)
Uzbekistan (UZ)	Venezuela (VE)	Vietnam (VN)
Vanuatu (VU)	Yemen (YE)	Mayotte (YT)
South Africa (ZA)		

Table 5: Countries included in the Language Bias analysis in Section 8. These countries do not have English as their primary language, as identified by GeoNames.
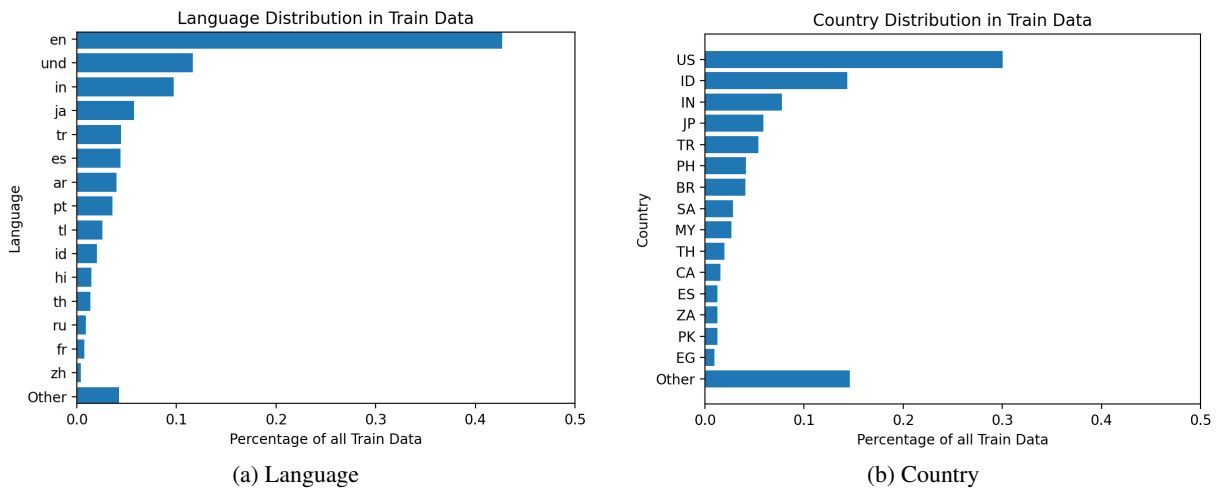
(a) Language

(b) Country

Figure 6: The distribution of languages and countries in the training dataset. For space, the top 15 from each category are shown individually, and the remaining are aggregated as "Other".

(a) F1 score with respect to country-level prediction for each country. 44 countries with 0.0 accuracy are not shown for space.

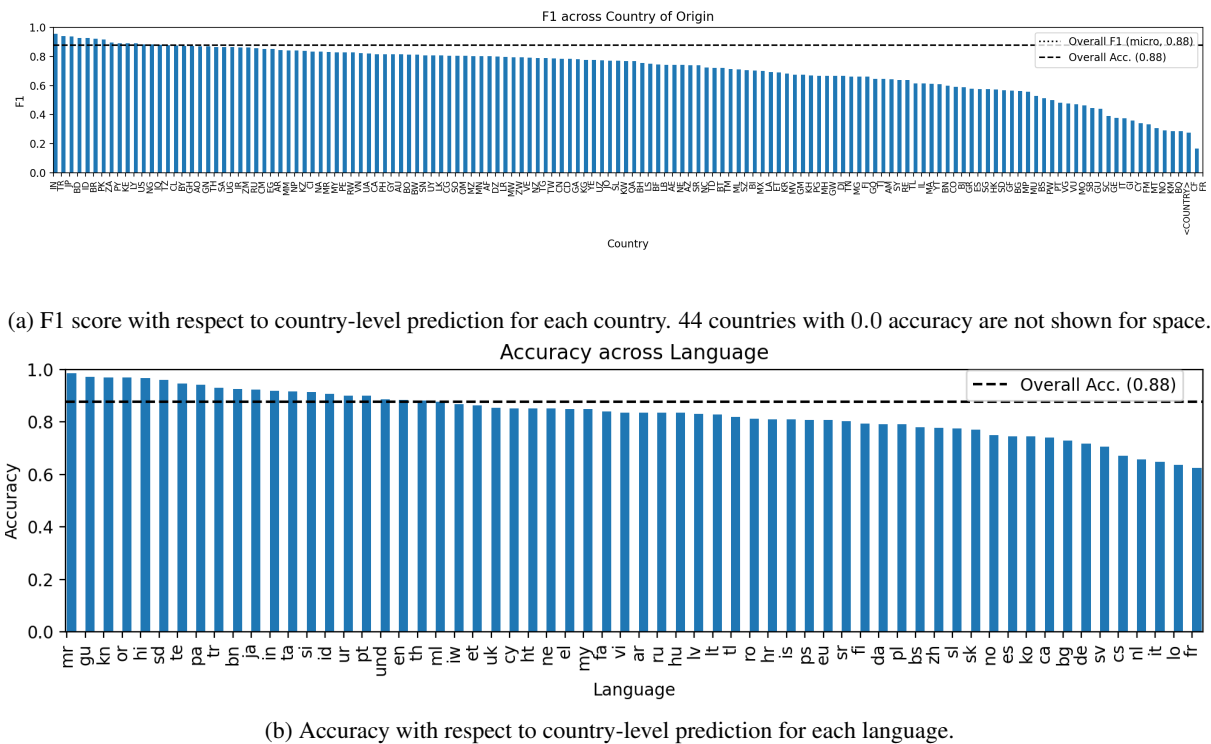(b) Accuracy with respect to country-level prediction for each language.

Figure 7: "Fairness" in GEO-SEQ2SEQ performance as measured by $mr_{country}$ across the 69 languages and 185 countries present in the TWITTER-PUG test set.