pigeo: A Python Geotagging Tool

Afshin Rahimi, Trevor Cohn, and **Timothy Baldwin**
Department of Computing and Information Systems
The University of Melbourne
arahimi@student.unimelb.edu.au
{t.cohn,tbaldwin}@unimelb.edu.au

Abstract

We present pigeo, a Python geolocation prediction tool that predicts a location for a given text input or Twitter user. We discuss the design, implementation and application of pigeo, and empirically evaluate it. pigeo is able to geolocate informal text and is a very useful tool for users who require a free and easy-to-use, yet accurate geolocation service based on pre-trained models. Additionally, users can train their own models easily using pigeo's API.

1 Introduction

Geolocation is the task of identifying a location for a user or document, and has applications in local search, recommender systems (Ho et al., 2012), targeted advertising (Lim and Datta, 2013), health monitoring (Paul et al., 2015), rapid disaster response (Ashktorab et al., 2014), and research with a regional restriction (Gutierrez et al., 2015), noting the potential privacy concerns associated with any such application (De Cristofaro et al., 2012). While primary service providers such as Twitter and Google are able to use metadata such as IP addresses, WiFi traces and direct access to a GPS signal to geolocate their users, this data is generally not available to third parties. This paper introduces a resource that can be used to geolocate users given textual messages generated by them, and the interactions between users encoded in those messages, focused particularly at Twitter data.

Both language use and social ties are geographically biased, and thus can be used to recover the location of a user or a document. Previous research has shown that the geographical bias in language use can be used in supervised text-based geolocation models, to learn associations between textual features and different regions based on large-scale collections of geotagged documents/tweets (Wing and Baldridge, 2011; Han et al., 2012; Maier and Gómez-Rodrıguez, 2014). Given an unseen piece of text or the text content of a user's timeline, the trained classifier can predict the most likely location(s) associated with the input.

Although social media services such as Twitter remove the geographical barrier for users to communicate, the majority of user interactions are still local (Backstrom et al., 2010). This geographical bias can be utilised to geolocate a user by analysing their social interactions. Based on the assumption that social interactions are more likely to be local, a user should be geographically close to their connections. The simplest approach to geolocation is to use the median location of a user's friends. Recent studies have shown that using both network and text information can improve the coverage and keep the predictions accurate simultaneously (Rahimi et al., 2015b).

Despite the widespread use of geolocation, most services are proprietary, overly-simplistic, or complicated to use. Supervised classification models often require huge amounts of geotagged data and large amounts of computing power to be trained. The performance is also heavily dependent on hyperparameter tuning, making the training procedure more challenging for end-users.

In this paper we introduce pigeo, a Python geolocation tool that has the following characteristics: (1) it comes with a pre-trained text-based model; (2) it is easy to use; (3) it has been tuned, benchmarked and proven to be accurate; (4) it supports both informal and formal text input; (5) it directly supports Twitter user geolocation; and (6) it has an easy-to-use RESTful API. pigeo is available at http://github.com/afshinrahimi/pigeo.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics—System Demonstrations, pages 127–132,
Berlin, Germany, August 7-12, 2016.

2 Background and Related Work

Prior work on geolocation falls broadly into two main categories: text-based and network-based methods. Both approaches use geotagged samples, and predict the location of an unseen document or user based on the trained model. Those approaches usually use GPS tags or user profile location fields as the ground truth both for training and evaluating the model. Geographical bias in language use is most evident for countries with different languages (e.g. Germany versus China), but also exists for countries which share the same languages (e.g. in the spelling of *centre* vs. *center* in British vs. American English). The linguistic geographical bias is not limited to these obvious cases, however, and includes the use of toponyms, names of people, sport teams, and dialectal terms. These differences in use of language can be captured in text-based geolocation models. Previous work have used topic models (Eisenstein et al., 2010) and supervised flat (Wing and Baldridge, 2011; Han et al., 2012; Han et al., 2013; Han et al., 2014; Rahimi et al., 2015b) and hierarchical (Wing and Baldridge, 2014) classification models. The main idea is to learn the geographical distribution of a given word across different locations from training data, and use it to predict a location for a new user.

Social ties have also been used for social media user geolocation. Backstrom et al. (2010) showed that Facebook users tend to interact more with nearby people ("location homophily"), and used this property to geolocate users based on the location of their friends, hence popularising network-based geolocation approaches. A graph is usually built based on Facebook friendship (Backstrom et al., 2010), Twitter follows (Rout et al., 2013), Twitter reciprocal @-mentions (Jurgens, 2013), or Twitter @-mentions (Rahimi et al., 2015b). The problem can also be formulated as classification (Rout et al., 2013) or regression over real-valued coordinates (Jurgens, 2013; Rahimi et al., 2015b). In classification models, the location label set can be pre-existing regional boundaries (e.g. countries or cities) or automatically generated through discretisation (e.g. a k-d tree). The label distribution of friends is then averaged and used as the location of a given user. In a regression model, the median coordinates of the friends of a user are often used for prediction.

Network-based models are generally more accurate than text-based models but can't geolocate users who don't interact with training users, which is the case for more than 30% of users in the case of reciprocal Twitter @-mentions (Jurgens et al., 2015). Relaxing the requirement on reciprocity increases the coverage of users, at the expense of lower accuracy (Rahimi et al., 2015a).

There are several other geolocation services and libraries which focus on Twitter, including `pigeo`TextGrounder (Wing and Baldridge, 2014) with a focus on targeted advertising, `pigeo`Carmen (Dredze et al., 2013) with a focus on help monitoring, `pigeo`MapAffil (Torvik, 2015) for affiliation mapping, and `pigeo`Tweedr (Ashktorab et al., 2014) for rapid disaster response. Many companies have their own proprietary geolocation service, which are either not available for public use or not open source. In `pigeo`, we provide trained a text-based classification model and network-based regression model for geolocation prediction, which has been benchmarked against standard datasets.

3 Methodology

`pigeo` uses two pre-trained models for geolocation: (1) `LR-WORLD` and (2) `LP-WORLD`. Both are trained on TWITTER-WORLD-EX, an extended version of the TWITTER-WORLD dataset (Han et al., 2012).

3.1 Data

We use TWITTER-WORLD-EX to train both the text-based classification and the network-based regression model. TWITTER-WORLD-EX is a Twitter dataset with global coverage (Han et al., 2012), comprising 1.3M geotagged users (188M tweets), of which 10K are held out for each of development and testing. The dataset contains predominantly English text, but also includes a rich variety of other languages. In TWITTER-WORLD, the location representation was cities, based on GEONAMES. For our purposes, we modify this to 930 clusters based on a k-d tree, to derive a smaller number of classes and remove class imbalance. Given that the dataset is about 5 years old, we expect the off-the-shelf performance to be degraded on newer tweets (Dredze et al., 2016), particularly in the case of the network-based model (Jurgens et al., 2015).

`LR-WORLD` is a text-based classification model trained over TWITTER-WORLD-EX. The train-

ing users of TWITTER-WORLD-EX are clustered into 930 regions with roughly the same number of users per region (about 2400), using a k-d tree. This results in many small regions/clusters in highly populated areas such as NYC, and a few large regions in sparsely-populated areas or areas with few Twitter users, such as the Sahara desert and China. The region IDs are then used as labels for all the users in that region. We use a bag-of-unigrams model of text with binary term frequency, inverse document frequency and l_2 normalisation of samples to create user vectors. Log loss is used with ElasticNet regularisation (90% l_1) as the cost function to train the model using stochastic gradient descent. Given an unseen text sample, one can vectorise the sample and use the classifier to predict a region/label or a probability distribution over regions. The predicted label(s) can be mapped to coordinates or locations.

The `LP-WORLD` model is a network-based regression model, also trained on TWITTER-WORLD-EX. An @-mention network is built over the dataset, and the real-valued coordinates of the training users are iteratively propagated to all the mentioned users. The location of each user is set to the weighted median latitude and weighted median longitude of all its connections. The edge weights are initially binary but are then normalised by dividing them by the product of the degree of the two corresponding nodes. The algorithm converges after 5 iterations. The predicted coordinates for all users are stored in a gzipped Python pickled dictionary for later use by `pigeo`. The Twitter user names are hashed by the MD5 algorithm for privacy reasons. The collision probability for MD5 hashing is very low and we didn't experience any collisions for our 7M nodes. Given an unseen Twitter user, the timeline of the user is downloaded and the @-mentions are extracted. The hashed content of each @-mention is looked up in the saved user-coordinate mapping to see if any predictions are available. The median latitude and longitude of geolocated @-mention connections are then predicted as the Twitter user location.

Although we experiment with the `LP-WORLD` model in this paper, we are unable to distribute it, due to Twitter's terms of service. It is possible, however, for a user to use `pigeo` to train their own network model by providing data in the format described in Section 4.

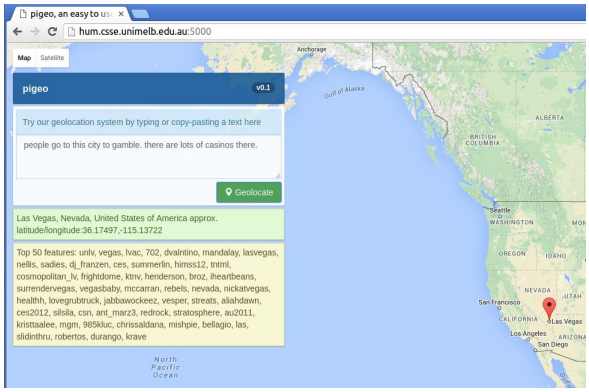

Figure 1: `pigeo`'s web interface. Given a piece of text or a single Twitter user, it geolocates it and returns the description and coordinates of the predicted location and its most important textual features in the model.

4 System Architecture

The main feature of `pigeo` is the ability to use the trained text-based classification network-based regression models that are distributed with the library, for geolocation of both text documents and Twitter users. Additionally, however, the library supports the training and storage of new text-based classification models. The `pigeo` tool is written in Python 2.7 and consists of: (1) the main `pigeo.py` script; (2) `params.py`, which stores the global parameters; (3) `twitterapi.py`, which uses `pigeo` to connect to Twitter; and (4) an `index.html` file, which is used by the web service. The tool returns a JSON string with fields such as `latitude`, `longitude`, `city` and `label_distribution`. `pigeo.py` packages all the main functions that are required by `pigeo`. It can be used in 3 modes: (1) Shell mode; (2) Web mode; and (3) Library mode.

Shell mode: Shell mode is activated as follows:

```
> python pigeo.py --mode shell
```

It takes an input text, geolocates it, and returns the result in JSON format. Shell mode uses the trained `LR-WORLD` model stored in `./models/world` and is best suited for testing `pigeo`.

Web mode: Web mode is activated by running:

```
> python pigeo.py --mode web
```

`pigeo` uses Flask, a lightweight Python web framework, to provide web access to end-users. The default host and port are 127.0.0.1 and 5000, respectively, which can be modified using the `--host` and `--port` options on the

```
import pigeo

# load the world model (default)
pigeo.load_model()

# geolocate a sentence
pigeo.geo("gamble casino city")

# geolocate a Twitter user
pigeo.geo('@POTUS')

# geolocate a list of texts
pigeo.geo(['city centre', 'city center'])
```

Figure 2: An illustration of Library mode

command line. When the service is running, the user can use the web service by opening `http://127.0.0.1:5000` via a web browser on their local machine shown, as illustrated in Figure 1. Alternatively, the users are able to use the `curl` command to geolocate a text or a Twitter user:

```
> curl 127.0.0.1:5000/geo?text='beach'
```

Library mode: `pigeo` can also be used as a library. This is the suggested way of using it if many documents are needed to be geolocated, because the batch functionality is only available in this mode. Note that running the `pigeo.geo` function in a loop is not as efficient as running it with a list argument (in **Batch mode**). The code snippet in Figure 2 shows how `pigeo` can be used in Library mode.

Twitter user geolocation: `pigeo` takes the user name of a Twitter user, crawls their timeline, and geolocates them on the basis of that data. This can be done in any of Shell, Web or Library modes, but requires an internet connection and valid Twitter authentication information (Twitter keys, tokens and secrets) which should be set in `twitterapi.py`.

Training a new model: Training a new model is possible in Library mode, using scikit-learn (Pedregosa et al., 2011) both for feature extraction and training the model. The training data consists of a list of text samples and a list of corresponding coordinates as a (latitude, longitude) tuple. Given the number of desired classes, `pigeo` discretises the training points and assigns a class to each training sample. The bag-of-unigram features are extracted using `TfidfVectorizer` and the model is

```
import pigeo

# train a model and save it in 'example'
pigeo.train_model(['text1', 'text2'],
[(lat1, lon1), (lat2, lon2)],
num_classes=2, model_dir='example')

# load and use the new model
pigeo.load_model(model_dir='example')
```

Figure 3: An illustration of training a model

```
import pigeo

# load lpworld
pigeo.load_lpworld()

# geolocate a Twitter user
pigeo.geo_lp('@potus')
```

Figure 4: An illustration of Twitter user geolocation using the network model

trained by `SGDClassifier` with log loss and ElasticNet regularisation. The end-user can manually tune the regularisation parameters using a held-out development set. The procedure for training is illustrated in Figure 3.

Network-based model: geolocation with the network-based model can be done similarly to `LR-WORLD`, but since the data is not recent, the results might not be as accurate as reported in Section 5. Given a Twitter user, the timeline is downloaded and the @-mentions are matched with the hashed user account names. The median location of the matched users is returned as the prediction. The procedure is illustrated in Figure 4.

4.1 Trained models

The trained `LR-WORLD` model distributed with `pigeo`, and we additionally document the `LP-WORLD`, in terms of the files, formats and characteristics of the model.

`LR-WORLD` contains 4 gzipped pickle files:

- `clf.pkl.gz` is a scikit-learn `SGDClassifier` instance trained on TWITTER-WORLD-EX, whose projection matrix is converted to a Scipy sparse matrix for scalability.
- `vectorizer.pkl.gz` is a scikit-learn `TfidfVectorizer` instance fitted to TWITTER-WORLD-EX which, given a text,

extracts the bag-of-unigram features with binary term frequency, inverse document frequency and l_2 normalisation of samples. Terms which occur in less than 10 documents are excluded.

- `coordinate_address.pkl.gz` is a dictionary that, given a (latitude, longitude) coordinate tuple, returns an address. It only covers the coordinates of the `LR-WORLD` classes and is based on geopy's OpenStreetMap API.

- `label_coordinate.pkl.gz` is a dictionary containing the classes/regions of the `LR-WORLD` model and their corresponding latitude/longitude tuple, which is the median of all the training points in that class.

`LP-WORLD` is made up of a single gzipped pickle file `userhash_coordinate.pkl.gz`, which is a dictionary of users mapped to predicted locations using label propagation over real-valued coordinates of TWITTER-WORLD-EX dataset. As we are unable to distribute this model, the user needs to provide it themselves.

5 Evaluation

We evaluate the performance of `LR-WORLD` and `LP-WORLD` model based on 3 evaluation measures used in previous research (Cheng et al., 2010; Eisenstein et al., 2010): the mean error (Mean), median error (Median), and the accuracy of geolocation within 161km of the actual location (Acc@161).

Note that lower values are better for Mean and Median, and higher values are better for Acc@161. The performance for the `LR-WORLD` and `LP-WORLD` models is shown in Table 1. Because there are no published results over TWITTER-WORLD-EX, we compared the performance of the models with previous work based on TWITTER-US (Wing and Baldridge, 2011).

6 Conclusion

We introduced `pigeo`, an easy-to-use, accurate Python geolocation tool which is able to geolocate both text and Twitter users based on two trained geolocation models: `LR-WORLD` and `LP-WORLD`. We described the implementation details of `pigeo`, and evaluated it on a standard Twitter geolocation dataset. It is our hope that `pigeo` will provides researchers with an accurate off-the-shelf baseline geolocation model for applications which require geolocation.

	Acc@161	Mean	Median
TWITTER-WORLD-EX dataset			
`LR-WORLD`	0.62	818	31
`LP-WORLD`	0.67	829	4
TWITTER-US dataset			
`LR-NA`	0.51	636	148
`LP-NA`	0.50	610	144
Wing and Baldridge (2014)	0.49	703	170

Table 1: The performance of the `LR-WORLD` text-based classification model and the `LP-WORLD` network-based regression model over the test set of TWITTER-WORLD-EX. The model performance over TWITTER-US is compared to previous work.

References

Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining Twitter to inform disaster response. In *Proceedings of The 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014)*, pages 354–358, University Park, USA.

Lars Backstrom, Eric Sun, and Cameron Marlow. 2010. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web (WWW 2010)*, pages 61–70, Raleigh, USA.

Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: a content-based approach to geo-locating Twitter users. In *Proceedings of the 19th ACM International Conference Information and Knowledge Management (CIKM 2010)*, pages 759–768, Toronto, Canada.

Emiliano De Cristofaro, Claudio Soriente, Gene Tsudik, and Albert Williams. 2012. Hummingbird: Privacy at the time of Twitter. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP)*, pages 285–299, San Francisco, USA.

Mark Dredze, Michael J Paul, Shane Bergsma, and Hieu Tran. 2013. Carmen: A twitter geolocation system with applications to public health. In *Proceedings of the AAAI 2013 Workshop on Expanding the Boundaries of Health Informatics Using AI (HIAI)*, pages 20–24, Bellevue, USA.

Mark Dredze, Miles Osborne, and Prabhanjan Kambadur. 2016. Geolocation for Twitter: Timing matters. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, San Diego, USA.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1277–1287, Boston, USA.

Carlos Gutierrez, Paulo Figuerias, Pedro Oliveira, Ruben Costa, and Ricardo Jardim-Goncalves. 2015. Twitter mining for traffic events detection. In *Science and Information Conference (SAI), 2015*, pages 371–378.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Geolocation prediction in social media data by finding location indicative words. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 1045–1062, Mumbai, India.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. A stacking-based approach to Twitter user geolocation prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013): System Demonstrations*, pages 7–12, Sofia, Bulgaria.

Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based Twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49:451–500.

Shen-Shyang Ho, Mike Lieberman, Pu Wang, and Hanan Samet. 2012. Mining future spatiotemporal events and their sentiment from online news articles for location-aware recommendation system. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 25–32, Redondo Beach, USA.

David Jurgens, Tyler Finethy, James McCorriston, Yi Tian Xu, and Derek Ruths. 2015. Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. In *Proceedings of the 9th International Conference on Weblogs and Social Media (ICWSM 2015)*, pages 188–197, Oxford, UK.

David Jurgens. 2013. That's what friends are for: Inferring location in online social media platforms based on social relationships. In *Proceedings of the 7th International Conference on Weblogs and Social Media (ICWSM 2013)*, pages 273–282, Boston, USA.

Kwan Hui Lim and Amitava Datta. 2013. A topological approach for detecting twitter communities with common interests. In *Ubiquitous Social Media Analysis*, pages 23–43. Springer.

Wolfgang Maier and Carlos Gómez-Rodrıguez. 2014. Language variety identification in Spanish tweets. In *Proceedings of the EMNLP2014 Workshop on Language Technology for Closely Related Languages and Language Variants*, pages 25–35, Doha, Qatar.

Michael J Paul, Mark Dredze, David A Broniatowski, and Nicholas Generous. 2015. Worldwide influenza surveillance through twitter. In *AAAI Workshop on the World Wide Web and Public Health Intelligence*, Austin, USA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2015a. Twitter user geolocation using a unified text and network prediction model. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics — 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 630–636, Beijing, China.

Afshin Rahimi, Duy Vu, Trevor Cohn, and Timothy Baldwin. 2015b. Exploiting text and network context for geolocation of social media users. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics — Human Language Technologies (NAACL HLT 2015)*, pages 1362–1367, Denver, USA.

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 (Hypertext 2013)*, pages 11–20, Paris, France.

Vetle I Torvik. 2015. Mapaffil: A bibliographic tool for mapping author affiliation strings to cities and their geocodes worldwide. *D-Lib Magazine*, 21(11):9.

Benjamin P Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (ACL-HLT 2011)*, pages 955–964, Portland, USA.

Benjamin P Wing and Jason Baldridge. 2014. Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 336–348, Doha, Qatar.