IMPROVING GEOCODING BY INCORPORATING GEOGRAPHICAL HIERARCHY AND ATTRIBUTES INTO TRANSFORMERS NETWORKS

by

Zeyu Zhang

A Dissertation Submitted to the Faculty of the

SCHOOL OF INFORMATION

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2023

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by: *Zeyu Zhang*, titled: *Improving Geocoding by Incorporating Geographical Hierarchy and Attributes into Transformers Networks*

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Steven Bethard — Date: Sep 13, 2023
Steven Bethard

Mihai Surdeanu (Sep 14, 2023 08:53 PDT) — Date: Sep 14, 2023
Mihai Surdeanu

Date: Sep 14, 2023
Clayton T Morrison

Xuan Lu (Sep 14, 2023 11:05 PDT) — Date: Sep 14, 2023
Xuan Lu

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Steven Bethard — Date: Sep 13, 2023
Steven Bethard
Dissertation Committee Chair
School of Information

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Steven Bethard, who has been the most important person in my graduate experience. As his student, I have felt extremely fortunate, and I could not have imagined having a better advisor and mentor for my Ph.D. studies. Throughout my Ph.D. journey, he has shown remarkable patience and understanding. No matter how many times my papers were rejected, he always encouraged me. Whenever I encountered difficulties, he provided effective solutions. In my heart, he is akin to a superhero, facing every challenge with a positive and uplifting spirit. If I have children in the future, I would certainly hope to raise them to be just like him.

Besides Steven, I would also like to extend my gratitude to my minor advisor, Mihai Surdeanu. I have worked closely with him for two years on a DARPA project. His expertise, understanding, and strategic guidance have been invaluable in my research, and I have learned a great deal from him. Additionally, I would like to thank the rest of my thesis committee: Clayton Morrison and Xuan Lu, for their insightful comments and encouragement, as well as the challenging questions which inspired me to broaden my research from various perspectives.

My very profound gratitude goes to my parents. They have not only provided me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation but have also been the very foundation of my existence. Their love, sacrifices, and unwavering belief in me brought me into this world and shaped the person I have become.

Finally, I must express my gratitude to all my collaborators: Dongfang Xu, Egoitz Laparra, and Zhengzhong Liang, for their support and cooperation in my research. Their insights and perspectives have been invaluable throughout this process. Last but not least, I would like to thank all of my friends: Yiyun Zhao, Xin Su, Zheng Tang, Jiacheng Zhang, Kadir Bulut Ozler, Sarah Stueve, and Xu Gao, for their friendship, support, and the good times we shared.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES – *Continued*

LIST OF TABLES – *Continued*

ABSTRACT

With the development of artificial intelligence, machines are empowering all aspects of people's lives. However, there are still many shortcomings in AI. For example, AI robots cannot accurately determine the place names in articles like humans. After all, there are too many places with the same name in the world. Therefore, Geocoding, the task of converting location mentions in text to structured spatial data, has recently seen progress thanks to a variety of new datasets, evaluation metrics, and machine-learning algorithms.

In this dissertation, I present empirical studies to explore four research questions: 1. Are classic information retrieval techniques competitive with modern neural approaches for toponym resolution? 2. Can transformer-based reranking improve over a strong candidate retrieval baseline? 3. Which kind of context is most effective for toponym resolution? 4. Is it better to approach toponym resolution as an ontology entry ranking paradigm or a geographic attribute prediction paradigm? Based on these questions, this dissertation contains four research projects, in which we first show that leveraging the better candidate generation, transformer-based reranking, and two-stage resolution can improve toponym resolution performance, and then introduce a new paradigm for toponym resoluton, which achieves a new state-of-the-art.

CHAPTER 1

Introduction

1.1 Motivation

Geocoding, also called toponym resolution or toponym disambiguation, is the subtask of geoparsing that disambiguates place names in text. The goal of geocoding is, given a textual mention of a location, to choose the corresponding geospatial coordinates, geospatial polygon, or entry in a geospatial database. Geocoders must handle place names (known as *toponyms*) that refer to more than one geographical location (e.g., *Paris* can refer to a town in the state of *Texas* in the *United States*, or the capital city of *France*), and geographical locations that may be referred to by more than one name (e.g., *Leeuwarden* and *Ljouwert* are two names for the same city in the Netherlands), as shown in fig. 1.1. Geocoding plays a critical role in tasks such as tracking the evolution and emergence of infectious diseases (Hay et al., 2013), analyzing and searching documents by geography (Bhargava, Spasojevic, and Hu, 2017), geospatial analysis of historical events (Tateosian et al., 2017), and disaster response mechanisms (Ashktorab et al., 2014; Bruijn et al., 2018)).

The field of geocoding, previously dominated by geographical information systems communities, has seen a recent surge in interest from the natural language processing community due to the interesting linguistic challenges this task presents. The four most

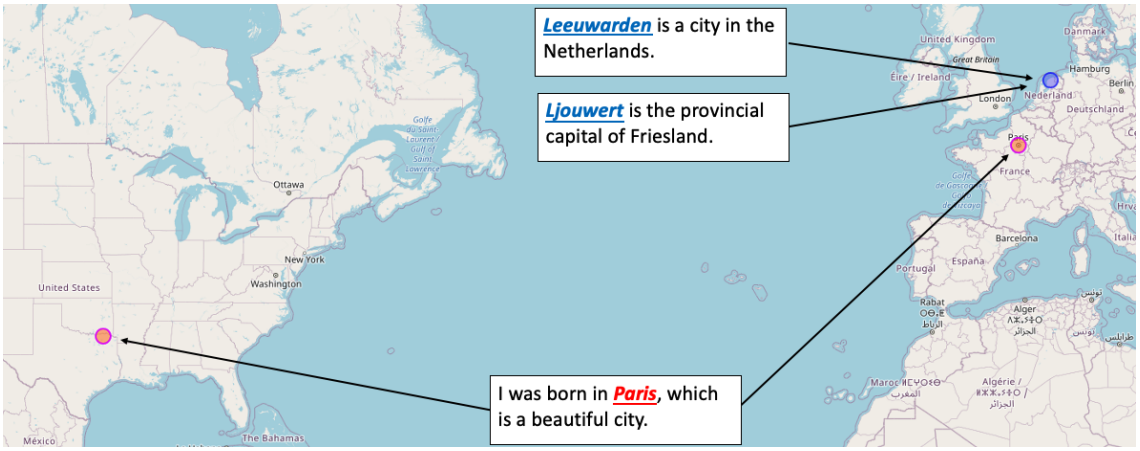

Figure 1.1: An illustrative example of geocoding challenges. One toponym (*Paris*) can refer to more than one geographical location (a town in the state of *Texas* in the *United States* or the capital city of *France* in *Europe*), and a geographical location may be referred to by more than one toponym (*Leeuwarden* and *Ljouwert* are two names for the same city in the Netherlands).

recent geocoding datasets (see table 2.1) were all published at venues in the ACL Anthology. And the recent ACL-SIGLEX sponsored SemEval 2019 Task 12: Toponym Resolution in Scientific Papers (Weissenbacher et al., 2019) resulted in several new natural language processing approaches to geocoding. The field has thus changed substantially since the most recent survey of geocoding (Gritta et al., 2017), including a doubling of the number of geocoding datasets, and the advent of modern neural network approaches to geocoding.

In this dissertation, we are interested in exploring the following research questions:

- Are classic information retrieval techniques competitive with modern neural approaches for toponym resolution?
- Can transformer-based reranking improve over a strong candidate retrieval baseline?
- Which kind of context is most effective for toponym resolution?

- Is it better to approach toponym resolution as an ontology entry ranking paradigm or a geographic attribute prediction paradigm?

1.2 Dissertation Outline

Our dissertation is organized as follows:

- In chapter 1, we present our motivations and research questions for our dissertation topic.
- In chapter 2, we will describe the related works and background for toponym resolution, including geocoding datasets and models.
- In chapter 3, we propose a candidate generator based on simple information retrieval techniques, that outperforms more complex neural models.
- In chapter 4, we proposed a new architecture for geocoding which is the first application of pre-trained transformers for toponym resolution and achieves state-of-the-art performance, outperforming prior work by large margins on toponym resolution corpora.
- In chapter 5, we propose a new two-stage resolution method to provide a simple and effective new approach to incorporating document-level context for geocoding.
- In chapter 6, we introduce a new paradigm for geocoding, where the machine learning algorithm encodes only the location mention, not the ontology.

- In chapter 7, we discuss our future work on toponym resolution.

CHAPTER 2

A Survey on Geocoding: Algorithms and Datasets for Toponym Resolution

Submitted to LRE

2.1 Chapter Overview

This chapter surveys currently available datasets, evaluation metrics, and algorithms for geocoding. In the remainder of this chapter, we first highlight some previous geocoding surveys (section 2.2) and explain the scope of the current survey (section 2.3). We then categorize the features of recent geocoding datasets (section 2.5), compare different choices for geocoding evaluation metrics (section 2.6), and break down the different types of features and architectures used by geocoding systems (section 2.7). We conclude with a discussion of where the field should head next (section 2.8).

2.2 Related Works

An early formal survey of geocoding is Leidner (2007), which distinguished finding place names (known as *geotagging* or *toponym recognition*) from linking place names to databases (known as *geocoding* or *toponym resolution*). They found that most geocoding methods were based on combining natural language processing techniques, such as lexical

string matching or word sense matching, with geographic heuristics, such as spatial-distance minimum and population maximum. Most geocoders studied in this thesis were rule-based.

Monteiro, Davis Jr, and Fonseca (2016) surveyed work on predicting document-level geographic scope, which often includes mention-level geocoding as one of its steps. Most of this survey focused on the document-level task, but the geocoding section found techniques similar to those found by Leidner (2007).

Gritta et al. (2017) reviewed both geotagging and geocoding, and proposed a new dataset, WikToR. The survey portion of this article compared datasets for geoparsing, explored heuristics of rule-based and feature-based machine learning-based geocoders, summarized evaluation metrics, and classified common errors from several geocoders (misspellings, case sensitivity, processing fictional and historical text presents, etc.). Gritta et al. (2017) concluded that future geoparsers would need to utilize semantics and context, not just syntax and word forms as the geocoders of the time.

Leidner (2021) reviewed many geospatial information processing tasks, but discussed only two geocoding systems in its section on geocoding.

Geocoding research since these previous surveys has changed in several important ways, as will be described in the remainder of this chapter. Most notably, new datasets and evaluation metrics are enabling new polygon-based views of the problem, and deep learning methods are offering new algorithms and new approaches for geocoding.

2.3 Article Inclusion Criteria

We focus on the geocoding problem, where mentions of place names are resolved to database entries or polygons. We thus searched the Google Scholar and Semantic Scholar search engines for papers matching any of the keyword queries: *geocoding*, *geoparsing*, *geolocation*, *toponym resolution*, *toponym disambiguation*, or *spatial information extraxtion*. From the results, we excluded articles that described tasks other than mention-level geocoding, for example:

- matching an entire document or microblog post to a single location (Luo et al., 2020; Hoang and Mothe, 2018; Kumar and Singh, 2019; Lee et al., 2015; Melo and Martins, 2017), as in geographic document retrieval and classification (Gey et al., 2005; Adams and McKenzie, 2018)

- matching typonyms to each other within a geographical database (Santos et al., 2018)

We also excluded papers published before 2010 (e.g., Smith and Crane, 2001), as they have been covered thoroughly by prior surveys.

In total, we reviewed more than 60 papers and included more than 30 of them in this survey.

2.4 Overview of the Survey

The survey is divided into three parts: geocoding datasets, geocoding evaluation metrics, and geocoding systems. In each part, we break down the relevant research to reveal the most common features shared across different research efforts and analyze the challenges and opportunities presented.

For geocoding datasets, we find that recent advances have led to an increased variety of domains, while the available geographic databases and geospatial label types have changed little. GeoNames remains the dominant geographic database, and point-based labels dominate over polygons. The availability of free polygon data on OpenStreetMap presents an opportunity to create new datasets that emphasize polygons over points.

For evaluation metrics, median error distance is preferred over mean error distance, and area under the curve of geocoding error distances (AUC) is favored over Accuracy@161 km. Yet these point-based metrics ignore the size and shape of geographic locations, while polygon-based metrics represent an opportunity to more carefully evaluate geocoding systems.

For geocoding systems, features like string matching and population are included in most systems regardless of whether they treat the problem as ranking or classification or whether they use deep neural networks or more traditional machine learning algorithms. Variability in selection of evaluation datasets makes direct comparison across systems difficult, but several systems have reported results on the LGL, WikTOR, GeoVirus, and WOTR datasets. These results generally show that deep neural network models

outperform more traditional machine learning algorithms. The neural network models typically incorporate fewer features (e.g., having limited notion of spatial distance), thus there is an opportunity to design deep learning architectures that can incorporate such features.

The remainder of this survey elaborates on these findings in detail.

2.5 Geocoding Datasets

Many geocoding corpora have been proposed, drawn from different domains, linking to different geographic databases, with different forms of geocoding labels, and with varying sizes in terms of both articles/messages and toponyms. Table 2.1 cites and summarizes these datasets, and the following sections walk through some of the dimensions over which the datasets vary.

Corpus	Domain	Geographic Database	Label Type	Articles / Messages	Toponyms
ACS (Mani et al., 2010)	News	GeoNames	Point	428	4783
LGL (Lieberman, Samet, and Sankaranarayanan, 2010)	News	GeoNames	Point & GeoNamesID	588	4793
CLUST (Lieberman and Samet, 2011)	News	GeoNames	Point & GeoNamesID	1082	11564
TUD-Loc-2013 (Katz and Schill, 2013)	Web	GeoNames	Point & GeoNamesID	152	3814
ZG (Zhang and Gelernter, 2014)	Twitter	GeoNames	Point & GeoNamesID	956	1393
WOTR (DeLozier et al., 2016)	Historical	OpenStreetMap	Point & Polygon	9653	10380
CLDW (Rayson et al., 2017)	Writing	Unlock	Point	80	-
WikTOR (Gritta et al., 2017)	Wikipedia	GeoNames	Point	5000	25000
Prussian (Ardanuy and Sporleder, 2017)	Historical	GeoNames	Point	N/A	1529
Belgian (Ardanuy and Sporleder, 2017)	Historical	GeoNames	Point	N/A	544
Antilles (Ardanuy and Sporleder, 2017)	Historical	GeoNames	Point	N/A	301
EastIndies (Ardanuy and Sporleder, 2017)	Historical	GeoNames	Point	N/A	210
DRegional (Ardanuy and Sporleder, 2017)	Historical	GeoNames	Point	N/A	1037
TR-NEWS (Kamalloo and Rafiei, 2018)	Historical	GeoNames	Point & GeoNamesID	118	1274
GeoCorpora (Wallgrün et al., 2018)	Twitter	GeoNames	Point & GeoNamesID	211	2966
GeoVirus (Gritta, Pilehvar, and Collier, 2018)	News	GeoNames	Point	229	2167
GeoWebNews (Gritta, Pilehvar, and Collier, 2019)	News	GeoNames	Point & GeoNamesID	200	5121
SemEval-2019-12 (Weissenbacher et al., 2019)	Scientific	GeoNames	Point & GeoNamesID	150	8360
GeoCoDe (Laparra and Bethard, 2020)	Wikipedia	OpenStreetMap	Polygon	360187	360187
TopRes19th (Ardanuy et al., 2022)	News	Wikipedia	Point	343	3364

Table 2.1: Summary of geocoding datasets covered by this survey, sorted by year of creation.

2.5.1 Domains

The news domain is the most common target for geocoding corpora, covering sources like broadcast conversation, broadcast news, news magazines, and newspapers. Examples include the ACE 2005 English SpatialML Annotations (ACS), the Local Global Lexicon (LGL), CLUST, TR-NEWS, GeoVirus, GeoWebNews, and TopRes19th. Though all these datasets include news text, they vary in what toponyms are included. For example, LGL is based on local and small U.S. news sources with most toponyms smaller than a U.S. state, while GeoVirus focuses on news about global disease outbreaks and epidemics with larger, often country-level, toponyms.

Web text is also a common target for geocoding corpora. Wikipedia Toponym Retrieval (WikToR) and GeoCoDe are both based on Wikipedia pages. ACS, mentioned above, also includes newsgroup and weblog data. And social media, specifically Twitter, is the target for ZG and GeoCorpora. TUD-Loc-2013 contains a variety of webpages including news articles and blogs. These corpora vary as widely as the internet text upon which they are based. For example, GeoCoDe and WikToR include the first paragraphs of Wikipedia articles, while ZG and GeoCorpora contain Twitter messages with place names that were highly ambiguous and mostly unambiguous, respectively.

Other geocoding domains are less common, but have included areas such as historical documents and scientific journal articles. The Official Records of the War of the Rebellion (WOTR) corpus annotates historical toponyms of the U.S. Civil War. Ardanuy and Sporleder (2017) created 5 historical multi-lingual datesets based on national, regional,

local, and colonial historical newspapers. CLDW contains historical writings about the English Lake District in the early seventeenth and early twentieth centuries. The SemEval-2019 Task 12 dataset is based on scientific journal papers from PubMed Central[1].

2.5.2 Geographic databases

All geocoding corpora rely on some database of geographic knowledge, sometimes also called a gazetteer or ontology. Such a database includes canonical names for places along with their geographic attributes such as latitude/longitude or geospatial polygon, and may include other information, such as population or type of place.

Most geocoding corpora have used GeoNames[2] as their geographic database, including ACS, LGL, CLUST, ZG, WikToR, TR-NEWS, GeoCorpora, GeoVirus, GeoWebNews, and SemEval-2019-12. GeoNames is a crowdsourced database of geospatial locations, with almost 7 million entries and a variety of information such as feature type (country, city, river, mountain, etc.), population, elevation, and positions within a political geographic hierarchy. The freely available version of GeoNames contains only a (latitude, longitude) point for each location, with the polygons only available with a premium data subscription, so most corpora based on GeoNames do not use geospatial polygons.

Geocoding corpora where recognizing geospatial polygons is important have typically turned to OpenStreetMap[3]. OpenStreetMap is another crowdsourced database of geospatial locations, which contains both (latitude, longitude) points and geospatial polygons for its

[1] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/
[2] https://www.geonames.org/
[3] https://www.openstreetmap.org/

locations. WOTR and GeoCoDe are based on OpenStreetMap.

Wikipedia and Unlock[4] have also been utilized although they are less common geographic databases. For example, in TopRes19th, the toponyms are annotated with the link to the corresponding Wikipedia entries, which can be used to obtain the geographic coordinates of the locations through their URLs.

2.5.3 Geospatial label types

Three different types of geospatial labels have been considered in geocoding corpora: database entries, (latitude, longitude) points, and polygons. All corpora except WTOR and GeoCoDe assign to each place name the (latitude, longitude) point that represents its geospatial center. Many of the GeoNames-based corpora (LGL, CLUST, TUD-Loc-2013, TR-NEWS, GeoCorpora, GeoWebNews, and SemEval-2019-12) also assign to each place name its GeoNames database ID. The WTOR corpus assigns to each place name a point or a polygon, and GeoCoDe assigns to each place name only a polygon. Figure A1 shows an example of a polygon annotation from GeoCoDe.

2.5.4 Challenges: geocoding datasets

While there have been significant improvements in geocoding datasets, the community has not successfully pivoted from point-based labels to the more precise representation of geographic areas as polygons. This is due primarily to the dominance of GeoNames

[4] `https://groups.inf.ed.ac.uk/geoparser/documentation/v1.1/epub/unlock.html`

Figure 2.1: The red-shaded area is the polygon label for *Biancavilla*, which is defined by the set of its boundary coordinates retrieved from OpenStreetMap.

as a geographic database. GeoNames provides polygons only for a fee, creating a barrier for individuals and organizations that that would like to pursue polygon-based geocoding research.

An additional challenge is associative toponyms, such as *Canadian* or *Russian*. Associative toponyms are included in many geocoding datasets, such as LGL, GWN, and TR-News, but the geographic databases include only literal toponyms (e.g., *Canada* or *Russia*). Resolving such toponyms will thus be more difficult, especially when their demonymic forms diverge from their names (e.g., *Netherlands* vs. *Dutch*).

2.5.5 Opportunities: geocoding datasets

An opportunity for future research on geocoding datasets is to pivot to polygon based labels, which can more faithfully represent complex regions. OpenStreetMap, though used less widely in geocoding research to date, offers free polygon data, and thus provides an opportunity to design new polygon-based geocoding datasets that are not limited by GeoNames fees. Such datasets would allow the development of geocoding systems that

better reflect the geography of the world.

Another opportunity in geocoding is to take advantage of the increased variety of domains now available, including historical documents, scientific documents, Wikipedia, and social media. Most work to date has focused on a single one of these domains, meaning there is a need to develop approaches to unify the various datasets, allowing more general and robust geocoding systems to be trained.

2.6 Geocoding Evaluation Metrics

Geocoding systems are evaluated on geocoding corpora using metrics that depend on the corpus's geospatial label type.

2.6.1 Database entry correctness metrics

When the target label type is a geospatial database entry ID, common evaluation metrics for multi-class classification tasks are applied. These metrics can also be used for corpora with (latitude, longitude) point labels by breaking the globe down into a discrete grid of geospatial tiles, and treating each geospatial tile like a database entry.

Accuracy is the number of place names where the system has predicted the correct database entry, divided by the number of place names. Accuracy is sometimes also called *Precision@1* or *P@1* when there is only one correct answer (as in the case for current geocoding datasets) and when the ranking-based system is turned into a classifier by taking

the top-ranked result as its prediction (the current standard for geocoding evaluation).

$$Accuracy = \frac{|\hat{U}|}{|U|}$$

where U is the set of human-annotated place names, $\hat{U}$ is the set of place names where the system's single prediction or top-1 ranked result is correct.

2.6.2 Point distance metrics

When the target label type is a (latitude, longitude) point, common evaluation metrics attempt to measure the distance between the system-predicted point and the human-annotated point.

Mean error distance calculates the mean over all predictions of the distance between each system-predicted and human-annotated point:

$$MeanErrorDist = \frac{\sum_{u \in U} dis(l_s(u), l_h(u))}{|U|}$$

where U is the set of all human-annotated place names, $l_s(u)$ is the system-predicted (latitude, longitude) point for place name u, $l_h(u)$ is the human-annotated (latitude, longitude) point for place name u, and dis is the distance between the two points on the surface of the globe.

Median Error Distance is defined in a similar way to mean error distance, but takes the median of the error distances rather than the mean.

Accuracy@k km/miles measures the fraction of system-predicted (latitude, longitude) points that were less than k km/miles away from the human-annotated (latitude, longitude) points. Formally:

$$Acc@k = \frac{|\{u|u \in U \wedge dis(l_s(u), l_h(u)) <= k\}|}{|U|}$$

where U, l_s, l_h, and dis are defined as above, and k is a hyper-parameter. A common choice for k is 161 km $\approx$ 100 miles (Cheng, Caverlee, and Lee, 2010).

Area Under the Curve (AUC) calculates the area under the curve of the distribution of geocoding error distances. A geocoding system is better if the area under the curve is smaller. Formally:

$$AUC = \ln \frac{ActualErrorDistance}{MaxPossibleErrors}$$

where $ActualErrorDistance$ is the area under the curve, and $MaxPossibleErrors$ is the farthest distance between two places on earth.

2.6.3 Polygon-based metrics

When the target label type is a polygon, evaluation metrics attempt to compare the overlap between the system-predicted polygon and the human-annotated polygon.

Polygon-based precision and recall were proposed by Laparra and Bethard (2020)

based on the intersection of system-predicted and human-annotated geometries. Formally:

$$Precision = \frac{1}{|S|} \sum_{i \in |S|} \frac{area(S_i \cap H_i)}{area(S_i)}$$

$$Recall = \frac{1}{|H|} \sum_{i \in |H|} \frac{area(S_i \cap H_i)}{area(H_i)}$$

where the S is the system-predicted set of polygons and H is the human-annotated set of polygons.

2.6.4 Challenges: geocoding evaluation metrics

Some challenges exist with specific metrics. A challenge of using mean error distance is its sensitivity to outliers: a few locations with large errors can skew the results and obscure the accuracy of the majority of locations. For instance, Gritta et al. (2017) found that roughly 20% of the places caused most of the errors. A challenge of using *Accuracy@k km/miles* is that it weights small and large errors equally, which may not properly reflect the expectations of users of geocoding systems.

A challenge for all point-based evaluation metrics is that locations are not points on the globe, but regions, and thus the point-based evaluation metrics that are currently popular do a poor job of measuring the actual shapes predicted by geocoding systems.

2.6.5 Opportunities: geocoding evaluation metrics

For the metrics with specific challenges, alternative metrics have been defined and could be used more widely in future research. Median error distance is similar to mean error distance, but is more robust to outliers. AUC is similar to *Accuracy@k km/miles*, but it gives more weight to smaller errors, which are often more significant than larger errors in practical applications (Jurgens et al., 2015).

A larger opportunity in geocoding evaluation is the application of polygon-based metrics. While to date such metrics have been applied only to one polygon-based dataset, polygon-based metrics could also be applied to datasets with database entry labels. This would give credit to geocoding systems when two or more database entries are equally applicable, such as a mention of "Dallas" which is ambiguous between city and county, and where the polygons of both choices overlap. By considering the overlap of polygons, polygon-based metrics could provide a more precise evaluation of geocoding performance in such cases.

GeoCoder	Method Type	Prediction Type	Database Independent	Polygon based
Edinburgh Parser (Grover et al., 2010)	Rule-based	Ranking	No	No
TGBRW-2010 (Tobin et al., 2010)	Rule-based	Ranking	No	No
MAC-2010 (Martins, Anastácio, and Calado, 2010)	Machine Learning	Ranking	No	No
IGeo (Lieberman, Samet, and Sankaranarayanan, 2010)	Rule-based	Ranking	No	No
LS-2011 (Lieberman and Samet, 2011)	Rule-based	Ranking	No	No
MG (Freire et al., 2011)	Machine Learning	Ranking	No	No
CLAVIN (Berico Technologies, 2012)	Rule-based	Ranking	No	No
LS-2012 (Lieberman and Samet, 2012)	Machine Learning	Ranking	No	No
GeoTxt (Karimzadeh et al., 2013)	Rule-based	Ranking	No	No
SPIDER (Speriosu and Baldridge, 2013a)	Machine Learning	Ranking	No	No
WISTR (Speriosu and Baldridge, 2013a)	Machine Learning	Ranking	No	No
TRAWL (Speriosu and Baldridge, 2013a)	Machine Learning	Ranking	No	No
CMU-Geolocator (Zhang and Gelernter, 2014)	Machine Learning	Ranking	No	No
SMFCM-2015 (Santos, Anastácio, and Martins, 2015)	Machine Learning	Ranking	No	No
Topocluster (DeLozier, Baldridge, and London, 2015)	Machine Learning	Classification	Yes	No
GeoSem (Ardanuy and Sporleder, 2017)	Machine Learning	Ranking	No	No
CBH, SHS (Kamalloo and Rafiei, 2018)	Machine Learning	Ranking	No	No
CamCoder (Gritta, Pilehvar, and Collier, 2018)	Deep Learning	Classification	No	No
HIS-2019 (Ardanuy et al., 2019)	Rule-based	Classification	No	No
DM_NLP (Wang et al., 2019a)	Machine Learning	Ranking	No	No
CME-2019 (Cardoso, Martins, and Estima, 2019)	Deep Learning	Classification&Regression	Yes	No
RS-2020 (Aldana-Bobadilla et al., 2020)	Rule-based	Classification	No	No
MLG (Kulkarni et al., 2020)	Deep Learning	Classification	Yes	No
LB-2020 (Laparra and Bethard, 2020)	Rule-based	Regression	Yes	Yes
DeezyMatch (Ardanuy et al., 2020a)	Deep Learning	Classification	No	No
Bi-LSTM (Fize, Moncla, and Martins, 2021)	Deep Learning	Regression	Yes	No
LGGeoCoder (Yan et al., 2021)	Deep Learning	Classification	No	No
TR-2022 (Cardoso, Martins, and Estima, 2022)	Deep Learning	Classification	Yes	No
Voting (Hu et al., 2023)	Deep Learning	Ensemble	No	No

Table 2.2: Summary of geocoding systems covered by this survey, sorted by year of creation.

2.7 Geocoding Systems

Table 2.2 summarizes the approaches of geocoders over the last decade. These models have different approaches to the prediction problem, ranging from ranking to classification to regression. They implement their predictive models with technology ranging from hand-constructed rules and heuristics, to feature-based machine-learning models, to deep learning (i.e., neural network) models that learn their own features.

2.7.1 Prediction types

Ranking is the most common approach to making geospatial predictions (Edinburgh Parser, TGBRW-2010, MAC-2010, IGeo, LS-2011, MG, CLAVIN, LS-2012, WISTR, GeoTxt, CMU-Geolocator, SMFCM-2015, GeoSem, CBH, SHS, DM_NLP, RS-2020). For example, most rule-based systems index their geospatial database with a search system like Lucene (`https://lucene.apache.org/`), and query that index to produce a ranked list of candidate database entries. This ranked list may be further re-ranked based on other features such as population or proximity. The type of scores used in re-ranking include binary classification score (MG, LS-2012, WISTR, CMU-Geolocator, CBH, SHS, DM_NLP), regression distance MAC-2010, the precision at the first position of the ranked list SMFCM-2015, and heuristics based on information in the geospatial database (Edinburgh Parser, TGBRW-2010, IGeo, LS-2011, CLAVIN, GeoTxt).

Classification is commonly used in making geospatial predictions when the Earth's surface has been discretized into tiny areas (Topocluster, CamCoder, HIS-2019, CME-

2019, MLG, DeezyMatch, TR-2022, LGGeoCoder). For example, *CamCoder* divides the Earth's surface into 7,823 tiles, and then changes the geospatial label of each toponym to the tile containing its coordinate. *CamCoder* then directly predicts one of 7823 classes for each toponym mention.

Regression is sometimes used for geospatial predictions when the label type is a (latitude, longitude) point or a polygon (CME-2019, LB-2020, Bi-LSTM). For example, LB-2020 predict a set of coordinates (i.e., a polygon) by applying operations over reference geometries, where the operations take sets of coordinates as inputs and produce sets of coordinates as outputs. Regression approaches to geocoding are rare because directly predicting coordinates over the entire surface of the Earth is challenging.

2.7.2 Features and heuristics

All geocoding systems combine string matching (exact string matching, Levenshtein distance, etc.) with other features and/or heuristics (population, words in nearby context, etc.). Details of such features are described in this section.

String match checks whether the place name matches any names in the geospatial database (Edinburgh Parser, TGBRW-2010, MAC-2010, IGeo, LS-2011, MG, CLAVIN, GeoTxt, CMU-Geolocator, SMFCM-2015, GeoSem, CBH, SHS, DM_NLP, HIS-2019, RS-2020, DeezyMatch, TR-2022, Bi-LSTM). String matching can be done exactly, or approximately with edit distance metrics like Levenshtein Distance. For example, GeoTxt calculates the Levenshtein Distance between the place name in the text and each candidate

entry from the geospatial database, and selects the candidate with the lowest edit distance.

Population looks at the size of the population associated with the candidate database entry, typically preferring more populous entries to less populous ones (Edinburgh Parser, TGBRW-2010, MAC-2010, IGeo, LS-2011, MG, LS-2012, CLAVIN, GeoTxt, CMU-Geolocator, SMFCM-2015, CBH, SHS, CamCoder, DM_NLP). For example, when the Edinburgh Parser geocodes the text *I love Paris*, it resolves *Paris* to PARIS, FRANCE instead of PARIS, TX, U.S. since the former has a greater population in the geospatial database.

Type of place looks at the geospatial feature type (country, city, river, populated place, facility, etc.) of a candidate database entry, typically preferring the more geographically prominent ones (Edinburgh Parser, TGBRW-2010, MAC-2010, IGeo, LS-2011, MG, CLAVIN, LS-2012, GeoTxt, TRAWL, CMU-Geolocator, SMFCM-2015, GeoSem, CBH, SHS, DM_NLP, TR-2022). For example, TGBRW-2010 prefers "populated places" to "facilities" such as farms and mines, when there are multiple candidate geospatial labels.

Words in the nearby context are used to disambiguate ambiguous place names (LS-2012, WISTR, CMU-Geolocator, SMFCM-2015, Topocluster, GeoSem, CBH, SHS, DM_NLP, CamCoder, CME-2019, MLG, LGGeoCoder, TR-2022). Ways of using context words range from simple to complex. For example, WISTR uses a context window of 20 words on each side of the target place name, aiming to benefit from location-oriented words such as *uptown* and *beach*. In contrast, CMU-Geolocator searches for common country and state names in other nearby location expressions, using these mostly unambiguous

place names to help resolve the target place name.

One sense per referent is a heuristic that assumes that all occurrences of a unique place name in the same document will refer to the same geographical database entry (Edinburgh Parser, TGBRW-2010, IGeo, LS-2011, GeoTxt, CBH, SHS, DM_NLP). For example, after each time that IGeo resolves a place name to a geospatial label, it propagates the same resolution to all identical place names in the remainder of the document.

Spatial minimality is a heuristic that assumes that place names in a text tend to refer to geospatial regions that are in close spatial proximity to each other (Edinburgh Parser, TGBRW-2010, IGeo, LS-2011, CLAVIN, SPIDER, Topocluster, GeoSem, CBH, SHS). For example, when IGeo geocodes the text *96 miles south of Phoenix, Arizona, just outside of Tucson*, it takes *Tucson* as an "anchor" toponym and resolves that first to get a target region. Then for *Phoenix*, it selects the geospatial label that is most geographically proximate to the target region.

2.7.3 Method types

Rule-based systems use hand-crafted rules and heuristics to predict a geospatial label for a place name (Edinburgh Parser, TGBRW-2010, IGeo, LS-2011, CLAVIN, GeoTxt, HIS-2019, RS-2020, LB-2020). The rule bases range in size from 2 to more than 200 rules, and rules may be formalized in rule grammars or defined more informally and provided as code. For example, IGeo uses a rule defined via code to identify place names in comma groups (e.g., "New York, Chicago and Los Angeles", all major cities in the U.S.), and then

resolves all toponyms by applying a heuristic uniformly across the entire group. As another example, LB-2020 uses 219 synchronous grammar rules to parse a target polygon from reference polygons by constructing a tree of geometric operators (e.g., BETWEEN(p_1, p_2) calculates the region between geolocation polygons p_1 and p_2).

Feature-based machine-learning systems use many of the same features and heuristics of rule-based systems, but provide these as input to a supervised classifier that makes the prediction of a geospatial label (MAC-2010, MG, LS-2012, WISTR, CMU-Geolocator, SMFCM-2015, Topocluster, GeoSem, CBH, SHS, DM_NLP). They typically operate in a two-step rank-then-rerank framework, where first an information retrieval system produces candidate geospatial labels, then a supervised machine-learning model produces a score for each candidate, and the candidates are reranked by these scores. Classification and ranking algorithms include logistic regression (WISTR), support vector machines (MAC-2010, CMU-Geolocator), random forests (MG, LS-2012), stacked LightGBMs (DM_NLP), and LambdaMART (SMFCM-2015). For example, MAC-2010 trains a support vector machine regression model using features such as the population and the number of alternative names for each candidate.

Deep learning systems often approach geocoding as a one-step classification problem by dividing the Earth's surface into an $N \times N$ grid, where the neural network attempts to map place names and their features to one of these $N \times N$ categories (CamCoder, CME-2019, MLG, DeezyMatch, Bi-LSTM, LGGeoCoder, TR-2022). Each system has a unique neural architecture for combining inputs to make predictions, typically based on

either convolutional neural networks (CNNs) or recurrent neural networks (RNNs).

CamCoder was the first deep learning based-geocoder. Its lexical model uses CNNs to create vectors representing context words (a window of 200 words, location mentions excluded), location mentions (context words excluded) and the target place name. Its geospatial model produces a vector using a geospatial label's population (from the database) as its prior probability. CamCoder concatenates the lexical and geospatial vectors for the final classification.

MLG is also a CNN-based geocoder, but it does not use population or other geospatial database information. It captures lexical features in a similar manner to CamCoder, but takes advantage of the S2 geometry (`https://s2geometry.io/`) to represent its geospatial output space in hierarchical grid-cells from coarse to fine-grained. MLG can predict the geospatial label of a place name at multiple S2 levels by mutually maximizing both precision and generalization of predictions.

CME-2019 and TR-2022 are RNN-based geocoders that uses HEALPix geometry (Gorski et al., 2005) to discretize the Earth's surface. It uses long short-term memory network with pre-trained Elmo embeddings (Peters et al., 2018) or the embeddings generated by the pre-trained BERT (Devlin et al., 2018) to create vectors representing the place name, local context (50 words around the place name), and larger context (paragraph or 500 words around the place name). The three vectors are concatenated and used to predict both the class of the HEALPix region and the coordinates of the centroid of the HEALPix class. This joint learning approach allows the two tasks to be mutually promoted and restricted.

	Accuracy@161km (↑)				Mean error distance (↓)			
GeoCoder	LGL	GeoVirus	WikTOR	WOTR	LGL	GeoVirus	WikTOR	WOTR
Edinburgh Parser (Grover et al., 2010)	76	78	42	-	8	5	31	-
CLAVIN (Berico Technologies, 2012)	71	79	16	-	13	6	43	-
GeoTxt (Karimzadeh et al., 2013)	68	79	18	-	14	6	47	-
SPIDER (Speriosu and Baldridge, 2013a)	68	-	-	67	12	-	-	4.8
SMFCM-2015 (Santos, Anastácio, and Martins, 2015)	71	-	-	-	8	-	-	-
Topocluster (DeLozier, Baldridge, and London, 2015)	63	-	26	-	12	-	38	-
GeoSem (Ardanuy and Sporleder, 2017)	-	-	-	68	-	-	-	4.5
CamCoder (Gritta, Pilehvar, and Collier, 2018)	76	82	65	-	7	3	11	-
CME-2019 (Cardoso, Martins, and Estima, 2019)	86	-	-	82	2.4	-	-	1.6
MLG (Kulkarni et al., 2020)	73	85	85	-	6.2	2.8	3.5	-
TR-2022 (Cardoso, Martins, and Estima, 2022)	91	-	-	87	2.2	-	-	1.1

Table 2.3: Reported results on LGL, WikToR, GeoVirus, and WOTR. For accuracy@161km, larger is better (↑). For mean error distance, smaller is better (↓).

2.7.4 Challenges: geocoding systems

One of the challenges in geocoding research is the lack of consistency in evaluation datasets used by different geocoders. While the LGL, WikTOR, GeoVirus, and WOTR datasets have been shared by multiple geocoders, there is still much variability in the choice of evaluation datasets. This can make it difficult to compare the performance of different geocoders and to draw meaningful conclusions from the results. We nevertheless present the partial comparison that is possible in table 2.3.

The table reveals a challenge for the neural network models: they are data hungry. The gains of neural network models over prior approaches are modest on smaller datasets, such as LGL and GeoVirus, and only become large on the larger datasets, such as WikTOR and WOTR. This need for large datasets may be due to the architectures themselves, or they may be a result of the simpler set of features input to neural network systems as compared to pre-neural-network systems.

2.7.5 Opportunities: geocoding systems

One opportunity for geocoding system research is to increase the size of the training datasets. This could be achieved by applying techniques like multi-task learning to train a single model using the variety of available geocoding datasets.

Another opportunity is to incorporate additional features into the deep learning models. For instance, document-level consistency features like *one sense per referent*, geospatial consistency features like *spatial minimality*, and additional database information beyond

population were used by geocoding systems before deep learning models. Designing neural architectures that can incorporate such features could yield performance gains not possible with the current feature sets.

2.8 Future Directions

A key direction of future research will be output representations. Many past geocoders focused on mapping place names to geospatial database entries (see column 4 of table 2.2). This was convenient, enabling fast resolution by applying standard information retrieval models to propose candidate entries from the database, but was limited by the simple types of matching that information retrieval systems could perform. Modern deep learning approaches to geocoding allow more complex matching of place names to geospatial locations, but typically rely on discretizing the Earth's surface into tiles to constrain the size of the network's output space. For the neural networks to achieve the fine-grained level of geocoding available in geocoding databases, they may need to consider hierarchical output spaces (e.g., Kulkarni et al., 2020) or compositional output spaces (e.g., Laparra and Bethard, 2020) that can express the necessary level of detail without exploding the output space.

Another key direction of future research will be the structure and evaluation of geocoding datasets. Most existing datasets and systems treat geocoding as a problem of identifying points rather than polygons (see column 4 of table 2.1 and column 5 of table 2.2). Yet the vast majority of real places in geospatial databases are complex polygons (as in fig. A1),

not simple points. More polygon-based datasets are needed, especially ones like GeoCoDe (Laparra and Bethard, 2020) that include complex descriptions of locations (e.g., *between the towns of Adrano and S. Maria di Licodia*) and not just explicit place names (e.g., *Paris*). The current state-of-the-art for complex geographical description geocoding is rule-based, but more polygon-based datasets will drive algorithmic research that can improve upon these rule-based systems with some of the insights gained from deep neural network approaches to explicit place name geocoding.

Finally, geocoding evaluation is still an open research area. Future research will likely extend some of the new polygon-based evaluation metrics. For example, using polygon precision and recall would give credit to a geocoding system that predicted the GeoNames entry *Nakhon Sawan* even if the annotated data used the entry *Changwat Nakhon Sawan*, since the polygons of these two place names are nearly identical.

2.9 Conclusion

After surveying a decade of work on geocoding, we have identifed several trends. First, combining contextual features with geospatial database information makes geocoders more powerful. Second, like much of NLP, geocoders have moved from rule-based systems to feature-based machine-learning systems to deep-learning systems. Third, the older rank-then-rerank approaches, combining information retrieval and supervised classification, are being replaced by direct classification approaches, where the Earth's surface is discretized into many small tiles. Finally, the field of geocoding is just beginning to look beyond a

point-based view of locations to a more realistic polygon-based view.

CHAPTER 3

Improving Toponym Resolution with Better Candidate Generation

Published to *SEM 2023 (Zhang and Bethard, 2023)

3.1 Chapter Overview

Approaches to geocoding include generate-and-rank systems that first use information retrieval systems to generate candidate entries and then rerank them with hand-engineered heuristics and/or supervised classifiers (e.g., Grover et al., 2010; Speriosu and Baldridge, 2013b; Wang et al., 2019b) and vector-space systems that use deep neural networks to encode place names and database entries as vectors and measure their similarity (e.g., Hosseini, Nanni, and Coll Ardanuy, 2020; Ardanuy et al., 2020b). In this chapter, we propose an effective candidate generator based on simple information retrieval techniques which can outperform more complex neural models, such as vector-space systems.

3.2 Proposed Method

We define the task of toponym resolution as follows. We are given an ontology or knowledge base with a set of entries $E = \{e_1, e_2, ..., e_{|E|}\}$. Each input is a text made up of sentences $T = \{t_1, t_2, \ldots, t_{|T|}\}$ and a list of location mentions $M = \{m_1, m_2, ..., m_{|M|}\}$

in the text. The goal is to find a mapping function $f(m_i) = e_j$ that maps each location mention in the text to its corresponding entry in the ontology.

We approach toponym resolution using a candidate generator followed by a candidate reranker. The candidate generator, $G(m, E) \rightarrow E_m$, takes a mention m and ontology E as input, and generates a list of candidate entries E_m, where $E_m \subseteq E$ and $|E_m| \ll |E|$. As the candidate generator must search a large ontology and produce only a short list of candidates, the goal for G will be high recall and high runtime efficiency. The candidate reranker, $R(m, E_m) \rightarrow \widehat{E_m}$, takes a mention m and the list of candidate ontology entries E_m, and sorts them by their relevance or importance to produce a new list, $\widehat{E_m}$. As the candidate ranker needs to work only with a short list of candidates, the goal for R will be high precision, especially at rank 1, with less of a focus on runtime efficiency.

Our candidate generator is inspired by prior work on geocoding in using information retrieval techniques to search for candidates in the ontology Grover et al., 2010; Berico Technologies, 2012. Accurate candidate generation is essential, since the generator's recall is the ceiling performance for the reranker. As we will see in section 3.4, our proposed candidate generator alone is competitive with complex end-to-end systems from prior work.

Our sieve-based approach, detailed in algorithm 1, tries searches ordered from least precise to most precise until we find ontology entries that match the location mention. Intuitively, our goal is for mentions like *Austria* to match the entry AUSTRIA [2782113] in GeoNames before it matches AUSTRALIA [2077456], but still allow a typo like *Australa*

Algorithm 1: Candidate generator.

Input: a location mention, m
a maximum number of candidates, k
the GeoNames ontology, E
Output: a list of candidate entries E_m

```
// Index ontology
I ← ∅
for e ∈ E do
    name ← CANONICALNAME(E, e)
    synonyms ← SYNONYMS(E, e)
    for n ∈ {name} ∪ synonyms do
        I ← I ∪ {CREATEDOCUMENT(n, e)}
    end
end
// Search for candidates
E_m ← ∅
for t ∈ { EXACT, FUZZY, CHARACTERNGRAM, TOKEN, ABBREVIATION,
  COUNTRYCODE } do
    E_m ← SEARCH(I, m, t)
    if E_m ≠ ∅ then
        break
    end
end
// Select top entries by population
E_m ← SORT(E_m, KEY = e → POPULATION(E, e))
return top k elements of E_m
```

to match AUSTRALIA [2077456].

We create one document in the index for each name n_e of an entry e in the GeoNames ontology. A location mention m is matched to a name n_e by attempting a search with each of the following matching strategies, in order:

EXACT m exactly matches (ignoring whitespace) the string n_e

FUZZY m is within a 2 character Levenshtein edit distance (ignoring whitespace) of n_e

CHARACTERNGRAM m has at least one character 3-gram overlap with n_e

TOKEN m has at least one token (according to the Lucene StandardAnalyzer) overlap with n_e

ABBREVIATION m exactly matches the capital letters of n_e

COUNTRYCODE e is a country and m exactly matches a e's country code

Once one of the searches has retrieved a list of matching names, we recover the ontology entry for each name, sort those ontology entries by their population in the GeoNames ontology, and return the k most populous ontology entries. This list, E_m is then the input to the candidate reranker.

3.3 Experiments

3.3.1 Datasets

We conduct experiments on three toponym resolution datasets. Local Global Lexicon (LGL; Lieberman, Samet, and Sankaranarayanan, 2010) was constructed from 588 news articles from local and small U.S. news sources. GeoWebNews (Gritta, Pilehvar, and Collier, 2019) was constructed from 200 articles from 200 globally distributed news sites. TR-News (Kamalloo and Rafiei, 2018) was constructed from 118 articles from various global and local news sources. As there are no standard publicly available splits for these datasets, we split each dataset into a train, development, and test set according to a 70%,

Dataset	Train		Dev.		Test	
	Topo.	Art.	Topo.	Art.	Topo.	Art.
LGL	3112	411	419	58	931	119
GeoWebNews	1641	140	281	20	477	40
TR-News	925	82	68	11	282	25

Table 3.1: Numbers of articles (Art.) and manually annotated toponyms (Topo.) in the train, development, and test splits of the toponym resolution corpora.

10% , and 20% ratio[1]. The statistics of all datasets are shown in table 3.1.

3.3.2 Database

Our datasets use GeoNames[2], a crowdsourced database of geospatial locations, with almost 7 million entries and a variety of information such as geographic coordinates (latitude and longitude), alternative names, feature type (country, city, river, mountain, etc.), population, elevation, and positions within a political geographic hierarchy. An example entry from GeoNames is shown in fig. 3.1.

3.3.3 Evaluation metrics

There is not yet agreement in the field of toponym resolution on a single evaluation metric. Therefore, we gather metrics from prior work and use all of them for evaluation.

Accuracy is the number of location mentions where the system predicted the correct database entry ID, divided by the number of location mentions. Higher is better, and a perfect model would have accuracy of 1.0.

[1] `https://github.com/clulab/geonorm`
[2] `https://www.geonames.org/`

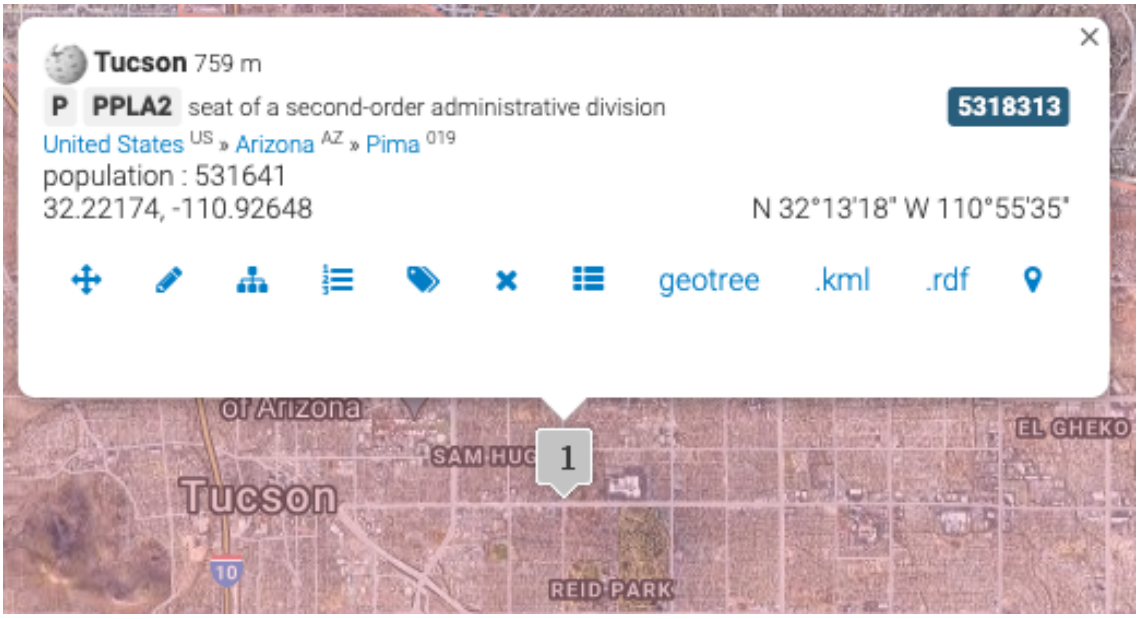

Figure 3.1: An entry for *Tucson* in GeoNames

Accuracy@161km measures the fraction of system-predicted (latitude, longitude) points that were less than 161 km (100 miles) away from the human-annotated (latitude, longitude) points. Higher is better, and a perfect model would have Accuracy@161km of 1.0.

Mean error distance calculates the mean over all predictions of the distance between each system-predicted and human-annotated (latitude, longitude) point. Lower is better, and a perfect model would have a mean error distance of 0.0.

Area Under the Curve calculates the area under the curve of the distribution of geocoding error distances. Lower is better, and a perfect model would have an area under the curve of 0.0.

3.3.4 Implementation details

We implement the candidate reranker with Lucene[3] v8.4.1 under Java 1.8. When indexing GeoNames, we also index countries under their adjectival forms in Wikipedia[4].

3.3.5 Systems

We compare to a variety of vector-space based geocoding systems:

DeezyMatch Hosseini, Nanni, and Coll Ardanuy (2020) introduced a vector-space approach that first pre-trains an LSTM-based classifier on GeoNames taking string pairs as input, and then fine-tunes the pair classifier on the target dataset. The trained DeezyMatch model compares mentions to database entries by generating vector representations for both and measuring their L2-norm distance or cosine similarity.

SAPBERT Liu et al. (2021) introduced a vector-space approach that pretrains a transformer network on the database using a self-alignment metric learning objective and online hard pairs mining to cluster synonyms of the same concept together and move different concepts further away. The pre-trained SAPBERT is then fine-tuned on the target dataset. SAPBERT was trained for the biomedical domain, but is easily retrained for other domains. We pretrain SAPBERT on GeoNames and finetune it on the toponym resolution datasets.

[3] https://lucene.apache.org/

[4] https://en.wikipedia.org/wiki/List_of_adjectival_and_demonymic_forms_for_countries_and_nations

Model	LGL (test)		GeoWebNews (test)		TR-News (test)	
	R@1	R@20	R@1	R@20	R@1	R@20
DeezyMatch	.172	.538	.262	.671	.206	.702
SAPBERT	.245	.742	.428	.746	.355	.780
GeoNorm (+gen, -rank)	**.606**	**.962**	**.694**	**.866**	**.716**	**.965**

Table 3.2: Performance of candidate generators on the test sets. R@1 is useful for measuring the accuracy of the candidate generator when used directly as a geocoder. R@20 is useful for estimating the ceiling performance of a top-20 reranker based on that candidate generator.

3.4 Results

We evaluate our context-free information retrieval based candidate generator, GeoNorm (+gen, -rank) comparing it to recent context-free candidate generators. Table 3.2 shows that our approach outperforms approaches from prior work by large margins, both in accuracy of the top entry (R@1) and whether the correct entry is in the top 20 (R@20).

3.5 Qualitative Analysis

Table 3.3 shows some qualitative analysis of errors that SAPBERT and our candidate generator with no reranking. Row 1 shows an example where SAPBERT fails but GeoNorm (+gen, -rank) succeeds by more effectively using population ranking. Row 2 shows an example where GeoNorm fails with a candidate generator alone by relying on population alone. But it looks we will succeed if we can jointly consider the name, population, and feature type information (ADM2 represents a county, PPLA2 represents a city).

	Example	Candidate				Rank	
		Name	Population	Feature Code	State	SAPBERT	GeonNorm (+gen, -rank)
1	*Ahlstrom said that the results found at the Washington Latin Public Charter School in Washington, D.C.*	**District of Columbia**	552433				1
		Washington County	147430			1	
2	*It was Los Angeles police officers she attempted to blow up.*	Los Angeles County	9818605	ADM2			1
		Los Angeles	3971883	PPLA2			2
		Los Angeles	125430	PPLA2			3
		Los Angeles	4217	PPL			4

Table 3.3: Examples of predictions from CamCoder and our candidate generator with no reranking (GeoNorm (+gen, -rank)). Target location mentions are underlined. Human annotated ontology entries are in bold.

3.6 Conclusion

We first propose our effective candidate generator based on simple lexical-based information retrieval techniques, which can achieve better performance than more complex neural-based vector space models. Our candidate generator considers only the text, so it may benefit by being refined by additional modules. Thus, the next step is to research rerankers to follow the candidate generator and consider additional geographical features.

CHAPTER 4

Improving Toponym Resolution with Transformer-based Reranking

Published to *SEM 2023 (Zhang and Bethard, 2023)

4.1 Chapter Overview

Recently, tile-classification systems that use deep neural networks to directly predict small tiles of the map rather than ontology entries has been proposed (e.g., Gritta, Pilehvar, and Collier, 2018; Cardoso, Martins, and Estima, 2019; Kulkarni et al., 2021). The deep neural network tile-classification approaches have been the most successful, but they do not naturally produce an ontology entry, which contains semantic metadata needed by users.

In chapter 3, our simple candidate generator has shown better performance than more complex vector-space systems that use deep neural networks(e.g., Hosseini, Nanni, and Coll Ardanuy, 2020; Ardanuy et al., 2020b). In this chapter, we propose a candidate reranker to rerank the short list returned by the candidate generator.

Our new architecture, GeoNorm (Candidate Generator + Candidate Reranker), shown in Figure 4.1, which builds on all of these lines of research: it uses pre-trained deep neural networks for the improved robustness in matching place names, while leveraging a generate-then-rank architecture to produce ontology entries as output.

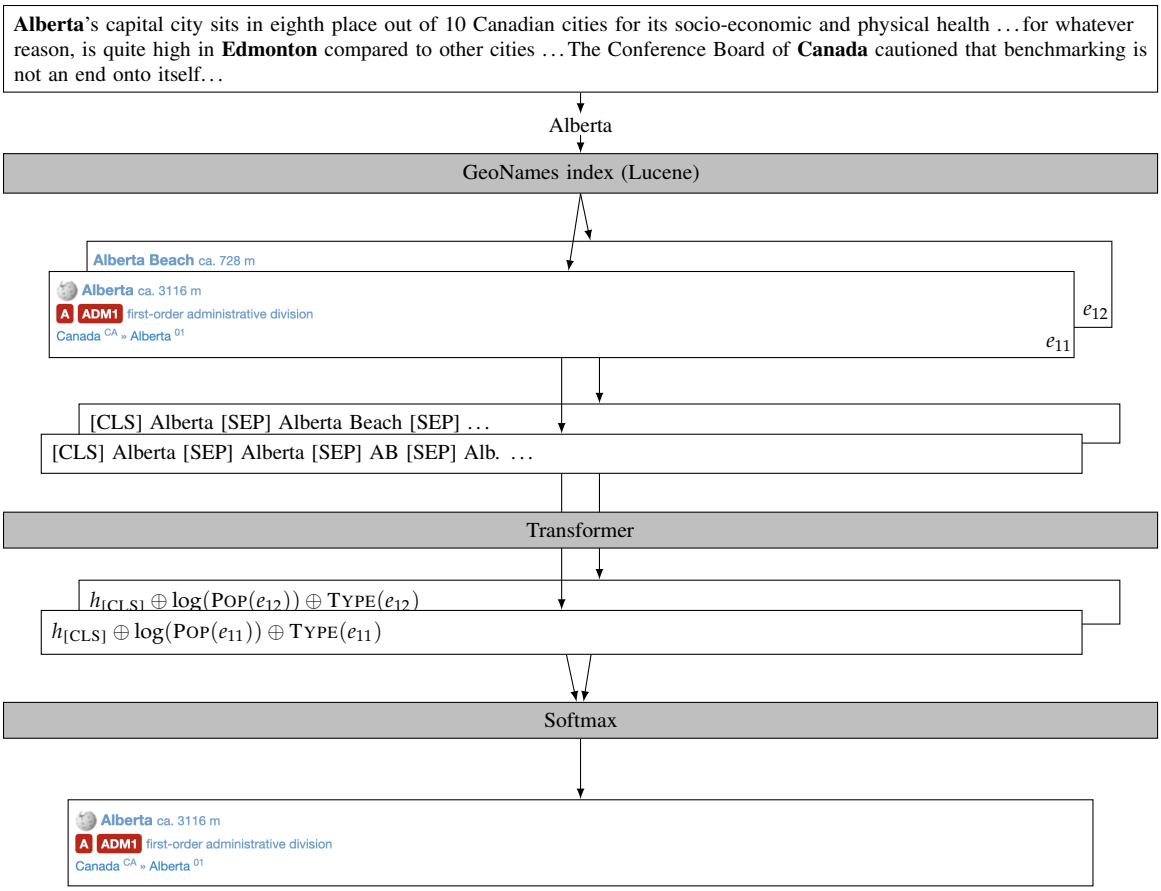

Figure 4.1: The architecture of our model, GeoNorm, applied to a sample text. The location mentions to be resolved are in bold.

Our reranker is the first application of pre-trained transformers for encoding location mentions and context for toponym resolution and our proposed architecture for geocoding achieves new state-of-the-art performance, outperforming prior work by large margins on toponym resolution corpora.

4.2 Proposed Method

Our candidate reranker is inspired by work on medical concept normalization Xu, Zhang, and Bethard, 2020; Ji, Wei, and Xu, 2020. The reranker takes a mention, m, and the list

of candidate entities from the candidate generator, E_m, encodes them with a transformer network, and uses these encoded representations to perform classification over the list to select the most probable entry. Formally, the model prediction, $\textsc{GeoNorm}(m, E_m) = \hat{e}$, is calculated as:

$$
\begin{aligned}
s^i &= \textsc{ToInput}(m, E_m^i) \\
\mathbf{A}^i &= \textsc{Transformer}(s^i) \\
\mathbf{b}^i &= \mathbf{A}_0^i \oplus \log(\textsc{Pop}(E, E_m^i)) \oplus \textsc{Type}(E, E_m^i) \\
c^i &= (\mathbf{b}^i \mathbf{W}_1^T)\mathbf{W}_2^T \\
\hat{\mathbf{y}} &= \text{softmax}(c^0 \oplus \ldots \oplus c^k)
\end{aligned}
$$

where:

- E_m^i is the i^{th} candidate entry for mention m
- $\textsc{ToInput}(m, e)$ produces a string of the form `[CLS]` m `[SEP]` $C(E, e)$ `[SEP]` $S(E, e)_1$ `[SEP]` ... `[SEP]` $S(E, e)_{|S(E,e)|}$ `[SEP]`, where $C(E, e)$ is the canonical name of e in the ontology, and $S(E, e)$ is the list of alternate names of e in the ontology.
- $\textsc{Transformer}(s)$ tokenizes the string s into word-pieces and produces contextualized embeddings for each of the word-pieces.
- $\mathbf{A}_0^i$ is the contexualized representation for the `[CLS]` token of candidate entry i's

input string

- $\textsc{Pop}(E, e)$ is the population of concept e in the ontology E
- $\textsc{Type}(E, e)$ is a one-hot vector identifying which of the T types in the ontology E the concept represents[1]
- $\oplus$ denotes vector concatenation
- $W_1 \in \mathbb{R}^{150\times(H+1+T)}$ and $W_2 \in \mathbb{R}^{1\times150}$ are learned weight matrices, where H is the transformer's hidden dimension
- $\hat{\mathbf{y}}$ is a probability distribution over the k entries proposed by the candidate generator

We represent the mention text + candidate entity synonyms with the contextualized representation of the `[CLS]` token, similar to applications of transformers to text classification. We include the population feature to allow the model to learn that locations in text are more likely to refer to high population than low population places (e.g., Paris, France vs. Paris, Texas, USA), and we take the logarithm of the population under the assumption that it is more important to capture the order of magnitude (e.g., thousands vs. millions) than the exact number. We include the type feature to allow the model to learn that locations in text are more likely to refer to some types of geographical features than others (e.g., San José, the capital of Costa Rica, vs. San José, the province).

[1] GeoNames has $T = 681$ types. For example, PPLC means *capital of a political entity*. Definitions for all types ("feature codes") are at `http://download.geonames.org/export/dump/featureCodes_en.txt`

The candidate reranker is trained with a standard classification loss:

$$L_R = \mathbf{y} \cdot log(\hat{\mathbf{y}})$$

where $\mathbf{y} \in \mathbb{R}^{|E_m|}$ is a one-hot vector representing the correct candidate entry.

4.3 Experiments

The datasets and evaluation metrics has been introduced in chapter 3

4.3.1 Implementation details

We implement the candidate reranker with the PyTorch[2] v1.7.0 APIs in Huggingface Transformers v2.11.0 (Wolf et al., 2020), using either `bert-base-uncased` or `bert-multilingual-uncased`. We train with the Adam optimizer, a learning rate of 1e-5, a maximum sequence length of 128 tokens, and a number of epochs of 30. We explored a small number of learning rates (1e-5, 1e-6, 5e-6) and epoch numbers (10, 20, 30, 40) on the development data. We use one Tesla V100 GPU with 32GB memory and a batch size of 8.

4.3.2 Systems

We compare to a variety of geocoding systems:

[2] `https://pytorch.org/`

Edinburgh Grover et al. (2010) introduced a rule-based extraction and disambiguation system that uses heuristics such as population count, spatial minimization, type, country, and some contextual information (containment, proximity, locality, clustering) to score, rank, and choose a candidate.

Mordecai Halterman (2017) introduced a generate-and-rank approach that uses Elasticsearch to generate candidates and neural networks based on word2vec (Mikolov et al., 2013) to rerank them. Its models are trained on proprietary data.

CamCoder Gritta, Pilehvar, and Collier (2018) introduced a tile-classification approach that combines a convolutional network over the target mention and 400 tokens of context with a population vector derived from location mentions in the context and populations from GeoNames. CamCoder predicts one of 7823 tiles of the earth's surface. The original CamCoder code, when querying GeoNames to construct its input population vector from location mentions in the context, assumes it has been given canonical names for those locations. Since canonical names are not known before locations have been resolved to entries in the ontology, we have CamCoder use mention strings instead of canonical names for querying GeoNames.

ReFinED Ayoola, Fisher, and Pierleoni (2022) introduced a vector-space approach for joint extraction and disambiguation of Wikipedia entities. One transformer network generates contextualized embeddings for tokens in the text, another generates embeddings

Model	LGL (dev)				GeoWebNews (dev)				TR-News (dev)			
	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
GeoNorm G	.594	.671	201	.289	.644	.858	73	.165	.677	.735	187	.242
GeoNorm GR	.802	.819	64	.141	.865	.925	39.5	.072	**.897**	**.912**	64.0	.081
GeoNorm GRP	.792	.819	68	.141	.861	.918	34.7	.072	.868	.882	65.7	.100
GeoNorm GRT	.807	**.828**	61	.134	.865	.915	31.9	.073	**.897**	**.912**	**42.7**	**.074**
GeoNorm GRPT	.797	.821	**57**	.140	**.886**	**.940**	**29.8**	**.060**	.882	.897	63.5	.090
GeoNorm GRPTM	**.814**	**.828**	60	**.132**	.879	.922	43.2	.072	.882	.897	65.0	.092

Table 4.1: Performance on the development sets. Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). The top score in each group is in bold, the second best score is underlined. Model features are indicated by the string of characters: G means the candidate generator was applied, R means a reranker was applied, P means the reranker included the population feature, T means the reranker included the type feature, M means the reranker was fine-tuned from `bert-multilingual-uncased` instead of `bert-base-uncased`.

for entries in the ontology, and tokens are matched to entries by comparing dot products over embeddings. ReFinED was trained on Wikipedia, but Wikipedia entries for place names have GeoNames IDs, and ReFinED can be further finetuned on the toponym resolution datasets.

4.3.3 Model selection

We performed model selection on the development sets as shown in table 5.2. All GeoNorm models that included a reranker (R) outperformed the candidate generator (G) alone. We explored the population (P) and type (T) features in models without context and fine-tune the reranker based on `bert-multilingual-uncased`, and found that they helped slightly on LGL and GeoWebNews but hurt slightly on TR-News. We thus selected the following models for evaluation: GeoNorm G and GeoNorm GRPTM.

	LGL (test)				GeoWebNews (test)				TR-News (test)			
Model	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
Edinburgh	.611	-	-	-	.738	-	-	-	.750	-	-	-
CamCoder	.580	.651	82	.288	.572	.665	155	.290	.660	.778	89	.196
Mordecai	.322	.375	926	.594	.291	.333	1072	.633	.472	.553	6558	.427
DeezyMatch	.172	.182	654	.704	.262	.323	537	.601	.206	.220	741	.705
SAPBERT	.245	.260	566	.630	.428	.499	357	.446	.355	.362	595	.568
ReFinED	.576	-	-	-	.658	-	-	-	.720	-	-	-
ReFinED (fine-tuned)	**.786**	-	-	-	.782	-	-	-	**.858**	-	-	-
GeoNorm G	.606	.685	119	.263	.694	.774	92	.194	.716	.812	95	.169
GeoNorm GRPTM	.761	**.785**	**59**	**.167**	**.788**	**.834**	**61**	**.131**	.798	**.816**	**89**	**.154**

Table 4.2: Performance on the test sets. Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). We do not report distance-based metrics for Edinburgh or ReFinED as these extraction+disambiguation systems do not make predictions for all mentions. The best performance on each dataset+metric is in bold. Model features are indicated by the string of characters: G means the candidate generator was applied, R means a reranker was applied, P means the reranker included the population feature, T means the reranker included the type feature.

4.4 Results

We evaluate our complete generate-and-rank system against other geocoders. We first perform model selection on the development set as described in section 4.3.3 to select four models to run on the test set: the candidate generator alone, the best generate-and-rank system. Table 4.2 shows that even without incorporating context, our generate-and-rank framework meets or exceeds the performance of almost all models from prior work. The exception is ReFinED, where our context-free model outperforms ReFinED out-of-the-box, but slightly underperforms our finetuned version of ReFinED.

4.5 Qualitative Analysis

The qualitative error analysis conducted for ReFinED, our standalone candidate generator (G), and our generate-and-rerank system (GR) is depicted in Table 4.3.

The first row presents an instance where ReFinED falls short, but GR prevails, attributing this success to the utilization of geospatial metadata, specifically population and feature type.

The second row introduces an instance where our standalone candidate generator (G) fails, but a context-free reranker brings about success. This improvement is achieved by a more comprehensive approach that takes into consideration the name, population, and feature type data simultaneously, rather than solely relying on population. Here, it's worth noting that "ADM2" signifies a county, whereas "PPLA2" implies a city.

The third row indicates a scenario where our GR model, devoid of context, still fails to deliver. The failure can be attributed to our inability to pinpoint the correct selection merely through the analysis of these geographical metadata. However, this hurdle could potentially be overcome with the utilization of contextual information.

4.6 Conclusion

We propose a new generate-and-rerank system, GeoNorm, for toponym resolution which consists of an information retrieval-based candidate generator, a BERT-based reranker that incorporates features important to toponym resolution such as population and type of location. Our GeoNorm has achieved state-of-the-art on many metrics on all of the

datasets. However, GeoNorm does not utilize any context information. To go further, we will try to incorporate the context into GeoNorm and also explore which kind of context is the most effective.

	Example	Candidate				Rank		
		Name	Pop.	Type	State	RF	G	GR
1	*The educational philosophy at the Washington Latin School in Alexandria is somewhat similar to Ahlstrom's previous endeavors.*	**Alexandria**	159467	PPLA2				1
		City of Alexandria	139966	ADM2		1		
2	*It was Los Angeles police officers she attempted to blow up.*	Los Angeles County	9818605	ADM2			1	2
		Los Angeles	3971883	PPLA2			2	1
		Los Angeles	125430	PPLA2			3	3
		Los Angeles	4217	PPL			4	4
3	*the Minnesota State Patrol urges motorists to drive with caution as flooding continues to affect area highways. Water over the roadway is currently affecting the following areas in Becker, Clay, and Douglas*	Clay County	221939		Missouri			1
		Clay County	190865		Florida			2
		Clay County	58999		Minnesota			3
		Clay County	26890		Indiana			4

Table 4.3: Examples of predictions from ReFinED (RF), our candidate generator alone (G) and our generate-and-rerank system without context (GR). Target location mentions are underlined. Human annotated ontology entries are in bold.

CHAPTER 5

Improving Toponym Resolution with Two-Stage Resolution

Published to *SEM 2023 (Zhang and Bethard, 2023)

5.1 Chapter Overview

In chapter 4, we proposed a new context-free architecture, GeoNorm, which leveraged a generate-then-rank architecture to produce ontology entries as output. In this chapter, we will incorporate the context information into GeoNorm and propose a two-stage approach that first resolves the less ambiguous countries, states, and counties, and then resolves the remaining location mentions, using the identified countries, states, and counties as context. The whole architecture is shown as Figure 5.1.

Our two-stage resolution provides a simple and effective new approach to incorporating document-level context for geocoding and the experimental results show that the document-level context can largely improve the toponym resolution performance.

5.2 Proposed Method

GeoNorm as introduced in chapter 4 only utilizes the mention and the candidate entry from ontology without incorporating context information. However, the text around a mention may provide clues (e.g., the context *Minnesota State Patrol urges motorists to*

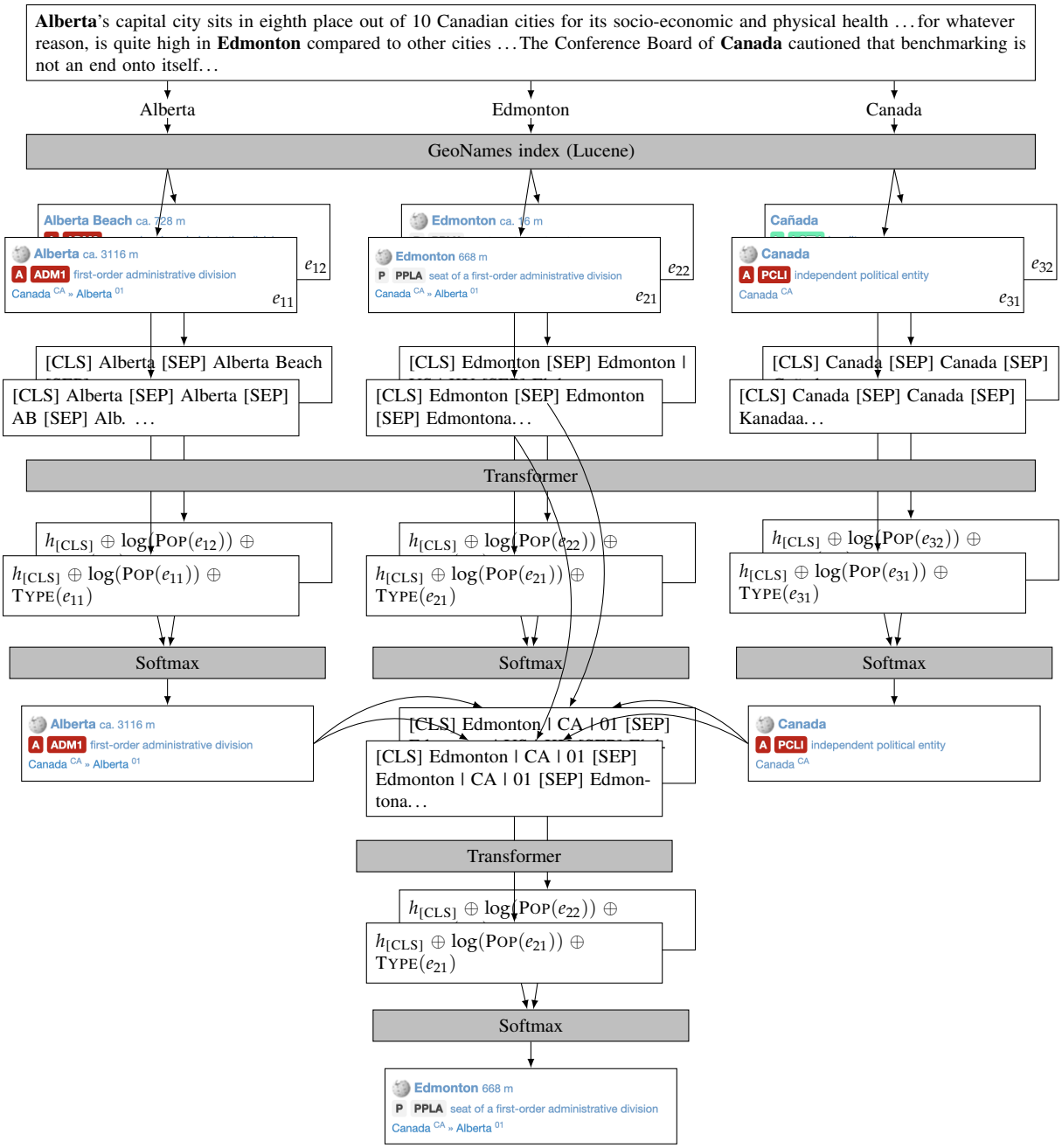

Figure 5.1: The architecture of our model, GeoNorm, applied to a sample text. The location mentions to be resolved are in bold.

drive with caution. . . in Becker, Clay, and Douglas suggests that *Clay* refers to Clay County, Minnesota, even though Clay County, Missouri is more populous). Thus, we consider two

Dataset	Precision				Recall			
	Country	State	County	Other	Country	State	County	Other
LGL	0.968	0.806	0.829	0.745	0.893	0.915	0.739	0.763
GWN	1.000	0.765	0.778	0.752	0.966	0.591	1.000	0.810
TR-News	1.000	1.000	0.000	0.830	1.000	1.000	0.000	0.830

Table 5.1: Precision and recall of GeoNorm (without context) on three geocoding development sets.

approaches to incorporating context.

context=*c*sent A simple approach is to take the c-sentence window surrounding the mention m and encode it with the the same transformer as was used to encode $m + e$. The contextualized representation of the c-sentence window's `[CLS]` token can then be concatenated into $\mathbf{b}$ alongside the other features. The 512 word-piece limit on the size of the transformer input means that this approach cannot incorporate the entire document.

context=2stage To include the full document context, we take advantage of the fact (demonstrated in table 5.1) that toponyms at the top of the hierarchy, like countries and states, can often be resolved precisely without context as they are less ambiguous. We thus propose Algorithm 2, a two-stage approach to geocoding. Lines 3-7 are the context-free stage, where GeoNorm is first applied to all location mentions. If the feature type of a predicted entry, $\texttt{TYPE}(e)$, is an administrative district 1-3 (i.e., the top of the geographic hierarchy: countries, states, or counties), then the prediction is accepted. Such predictions are converted to their administrative codes (e.g., *United States* $\rightarrow$ `US`) and added to the context. Lines 8-11 are the second stage, where the geocoding system is applied to all

Algorithm 2: Two-stage toponym resolution using document-level context.

Input: location mentions, M
GeoNames ontology, E

$\hat{R} \leftarrow \{\}$
$C \leftarrow \emptyset$
// Resolve toponyms without context
for $m \in M$ **do**
  $\hat{e} \leftarrow \text{GEONORM}(m, E)$
  if $\text{TYPE}(\hat{e}) \in \{\texttt{adm1}, \texttt{adm2}, \texttt{adm3}\}$ **then**
    $\hat{R}[m] \leftarrow \hat{e}$
    $C \leftarrow C \cup \{\text{CODE}(\hat{e})\}$
  end
end
// Resolve toponyms with context
$c \leftarrow$ `"|".join`(C)
for $m \in M$ **do**
  if $m \notin \hat{R}$ **then**
    $\hat{R}[m] \leftarrow \text{GEONORM}(m + c, E)$
  end
end
return $\hat{R}$

remaining location mentions but this time incorporating the collected context.

5.3 Experiments

The datasets, evaluation metrics and previous geocoding systems has been introduced in chapter 3.

5.3.1 Implementation details

We train with the Adam optimizer, a learning rate of 1e-5, a maximum sequence length of 128 tokens, and a number of epochs of 30. We explored a small number of learning rates

Model	LGL (dev)				GeoWebNews (dev)				TR-News (dev)			
	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
GeoNorm GRPTC1	.807	.823	55	.132	.865	.915	39.3	.075	.882	.882	110	.109
GeoNorm GRPTC3	.807	.816	65	.142	.868	.918	40.3	.073	.882	.897	64.9	.092
GeoNorm GRPTC5	.802	.814	68	.145	.865	.911	42.8	.078	.897	.912	64.0	.081
GeoNorm GRPTMC1	.816	.831	62	.133	.872	**.940**	**23.5**	**.057**	.882	.897	64.6	.090
GeoNorm GRPTMC3	.809	.833	59	.129	.875	.922	35.4	.073	.912	.927	40.6	.063
GeoNorm GRPTMC5	.807	.823	61	.137	.872	**.940**	29.4	.060	.868	.882	72.6	.103
GeoNorm GRPTMCD	**.885**	**.897**	**29**	**.079**	**.879**	.925	31.0	.065	**.971**	**.985**	**6.8**	**.010**

Table 5.2: Performance on the development sets. Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). The top score in each group is in bold, the second best score is underlined. Model features are indicated by the string of characters: G means the candidate generator was applied, R means a reranker was applied, P means the reranker included the population feature, T means the reranker included the type feature, M means the reranker was fine-tuned from `bert-multilingual-uncased` instead of `bert-base-uncased`, C1/C3/C5 means the reranker included 1/3/5 sentences of context, and CD means the reranker included the two-stage document-level context algorithm.

(1e-5, 1e-6, 5e-6) and epoch numbers (10, 20, 30, 40) on the development data. When training with context as *c*sent, we use four Tesla V100 GPU with 32GB memory and a batch size of 32. The total number of parameters in our model is 168M and the training time is about 3 hours. When training with context as 2stage, we use one Tesla V100 GPU with 32GB memory and a batch size of 8.

5.3.2 Model selection

We performed model selection on the development sets as shown in table 5.2. All GeoNorm models that included a reranker (R) outperformed the candidate generator (G) alone. We explored the population (P) and type (T) features in models without context, and found that they helped slightly on LGL and GeoWebNews but hurt slightly on TR-News. For

models with context, rerankers fine-tuned from `bert-multilingual-uncased` (M) slightly outperformed models fined-tuned from `bert-base-uncased`. Adding sentence level context (C1/C3/C5) to the rerankers helped on TR-News, but did not help on LGL or GeoWebNews. Applying the two-stage algorithm for document-level context led to large gains on LGL and TR-News, but did not help on GeoWebNews.

We thus selected the following models for evaluation: GeoNorm G, GeoNorm GRPT, and GeoNorm GRPTMCD.

5.4 Results

We evaluate our complete generate-and-rank system with context against other geocoders. We first perform model selection on the development set as described in section 5.3.2 to select four models to run on the test set: the candidate generator alone, the best generate-and-rank system with no context, and the best generate-and-rank system with context. Table 5.3 shows that our proposed GeoNorm model outperforms all prior work across all toponym resolution test sets on all metrics. Adding the novel two-stage document-level context yields large gains over the context free version of our model, and outperforms even the finetuned ReFinED. The final row the table shows the peformance of a model trained on the combined training data from all datasets, which we release for English geocoding under the Apache License v2.0, for off-the-shelf use at `https://github.com/clulab/geonorm`.

Model	LGL (test)				GeoWebNews (test)				TR-News (test)			
	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
Edinburgh	.611	-	-	-	.738	-	-	-	.750	-	-	-
CamCoder	.580	.651	82	.288	.572	.665	155	.290	.660	.778	89	.196
Mordecai	.322	.375	926	.594	.291	.333	1072	.633	.472	.553	6558	.427
DeezyMatch	.172	.182	654	.704	.262	.323	537	.601	.206	.220	741	.705
SAPBERT	.245	.260	566	.630	.428	.499	357	.446	.355	.362	595	.568
ReFinED	.576	-	-	-	.658	-	-	-	.720	-	-	-
ReFinED (fine-tuned)	.786	-	-	-	.782	-	-	-	.858	-	-	-
GeoNorm G	.606	.685	119	.263	.694	.774	92	.194	.716	.812	95	.169
GeoNorm GRPTM	.761	.785	59	.167	.788	.834	61	.131	.798	.816	89	.154
GeoNorm GRPTMC3	.759	.783	67	.166	.782	.832	60	.131	.777	.798	92	.166
GeoNorm GRPTMCD	**.807**	**.824**	**46**	**.135**	**.828**	**.862**	**55**	**.114**	**.918**	**.933**	**34**	**.057**
GeoNorm released	.799	.828	52	.136	.832	.876	54	.104	.897	.911	36	.073

Table 5.3: Performance on the test sets. Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). We do not report distance-based metrics for Edinburgh or ReFinED as these extraction+disambiguation systems do not make predictions for all mentions. The best performance on each dataset+metric is in bold (excluding the final model that was trained on more data). Model features are indicated by the string of characters: G means the candidate generator was applied, R means a reranker was applied, P means the reranker included the population feature, T means the reranker included the type feature, M means the reranker was fine-tuned from `bert-multilingual-uncased` instead of `bert-base-uncased`. CD means the reranker included the two-stage document-level context algorithm.

5.5 Qualitative Analysis

Table 5.4 shows some qualitative analysis of errors that ReFinED and different variants of GeoNorm made. Row 1 shows an example where ReFinED fails but GeoNorm succeeds, by more effectively using geospatial metadata such as population and feature type. Row 2 shows an example where GeoNorm fails without context but succeeds with context, by taking advantage of the *Minnesota* in the context to select the *Clay County* that would

otherwise seem implausible due to its lower population. Finally, row 3 shows an example where our best GeoNorm model still fails. The candidate generator includes the correct ontology entry in its top-k list, but neither the name, population, feature code, nor nearby context suggest the correct candidate. The global context includes toponyms from the same state, allowing the model with document context to move the correct answer up from rank 4 to rank 2. But fully addressing this issue would likely require predicting countries and states of toponyms in the text before resolving them.

5.6 Conclusion

We propose a novel two-stage resolution strategy that incorporates document-level context to GeoNorm for toponym resolution. We evaluate our GeoNorm with two-stage resolution against prior state-of-the-art, using multiple evaluation metrics and multiple datasets and show it can achieve new state-of-the-art performance on all datasets. However, according our qualitative analysis, we found that GeoNorm with two-stage resolution can not perform very well when there is no hierarchical locations in the document. Thus, how to effectively predict countries and states of toponyms when there is no hierarchical locations in the text will be our next research topic.

	Example	Candidate				Rank			
		Name	Pop.	Type	State	RF	G	GR	GRC3	GRCD
1	*The educational philosophy at the Washington Latin School in Alexandria is somewhat similar to Ahlstrom's previous endeavors.*	**Alexandria**	159467	PPLA2				1		1
		City of Alexandria	139966	ADM2		1				
2	*the Minnesota State Patrol urges motorists to drive with caution as flooding continues to affect area highways. Water over the roadway is currently affecting the following areas in Becker, Clay, and Douglas*	Clay County	221939		Missouri			1		4
		Clay County	190865		Florida			2		3
		Clay County	58999		Minnesota			3		1
		Clay County	26890		Indiana			4		2
3	*he writes, as do my efforts to insure New London is a safe community.*	New London County	274055	ADM2			1		3	4
		New London	27179	PPL			2		1	1
		New London	7172	PPL			3		2	3
		New London	1882	PPL			4		4	2

Table 5.4: Examples of predictions from ReFinED (RF), our candidate generator alone (G), our generate-and-rerank system without context (GR), our system with sentence context (GRC3), and our system with 2-stage document context (GRCD). Target location mentions are underlined. Human annotated ontology entries are in bold.

CHAPTER 6

Improving Toponym Resolution by Predicting Attributes to Constrain Geographical Ontology Entries

Submitted to EMNLP 2023

6.1 Chapter Overview

The traditional paradigm for geocoding systems is to train machine learning algorithms that encode both the location mention and the geographical ontology entries when predicting a label for the mention. For example, CamCoder's (Gritta, Pilehvar, and Collier, 2018) convolutional network encodes the location mention, the nearby context, and a population vector derived from the ontology entries, while encodes the location mention, the document level context, and the alternative names, location hierarchy, and population from the ontology. This common paradigm of using machine learning to encode both the location mention and the ontology entries requires the machine learning algorithm at prediction time to examine not only the text but also the ontology.

In this chapter, we propose an alternative prompt-based approach to geocoding, where the machine learning algorithm at prediction time needs to encode only the location mention and its context, not the ontology. Rather than directly predicting ontology entries, our machine learning algorithm predicts attributes of ontology entries, such as the enclosing

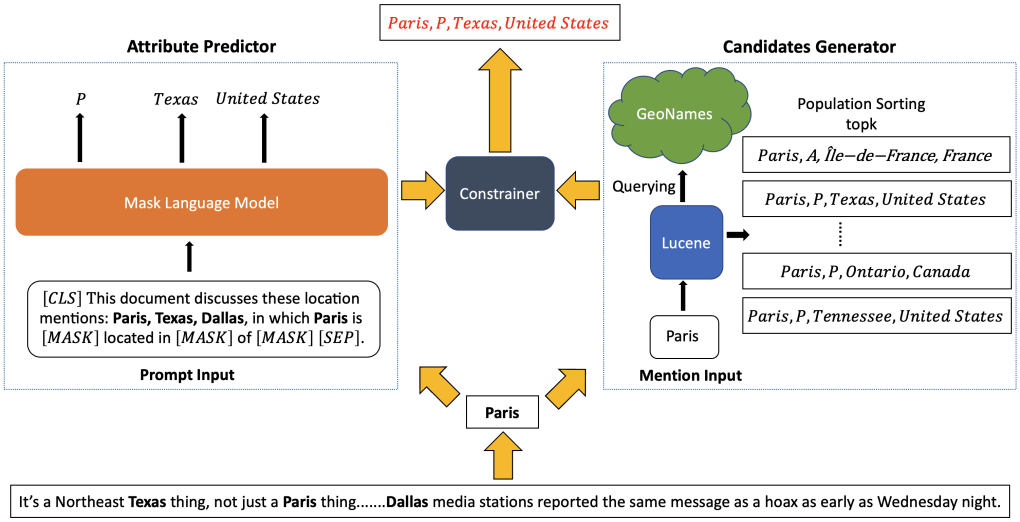

Figure 6.1: The architecture of our model: GEOgraphical normalization by Predicting Attributes to Constrain Ontology Entries (GeoPLACE). The figure shows how GeoPLACE normalizes a mention of *Paris*.

country and state. For example, our approach would predict that *Paris* in a document about Texas would have the attributes *"a Populated Place located in Texas in the United States."* The constraints implied by these predictions then deterministically filter the ontology entries. Our novel architecture, GEOgraphical normalization by Predicting Location Attributes to Constrain ontology Entries (GeoPLACE) is illustrated in Figure 6.1.

Our work makes the following contributions:

- We introduce a new paradigm for geocoding, where the machine learning algorithm encodes only the location mention, not the ontology.

- We design a transformer network for predicting the country, state, and feature type of a location mention, based on a masked language modeling objective and a novel prompt including all location mentions in the document.

- We demonstrate that the algorithm for predicting country, state, and feature type can be further improved by pre-training on synthetic data generated from the ontology.
- We introduce a novel deterministic algorithm that constrains the list of ontology entries using the predicted country, state, and feature type.
- Our proposed approach achieves new state-of-the-art performance on multiple datasets.

6.2 Proposed Methods

The problem of geocoding can be formalized as defining a function $f(m|T, M, E) = \hat{e}$ where T is the text of a document, M is the set of location mentions in the document, E is the set of geographical database entries, $m \in M$ is the mention under consideration, and $\hat{e} \in E$ is the entry predicted by f for m. In our paradigm for geocoding, we formulate f to first predict the country, state, and feature type of m, next query the ontology with m to find candidate entries, then select the entry that violates the fewest constraints implied by the predicted attributes as the prediction $\hat{e}$. Formally:

$$\hat{C}_m, \hat{S}_m, \hat{F}_m = \textsc{AttributePredictor}(m, M)$$

$$\hat{E} = \textsc{CandidateGenerator}(m, E)$$

$$f(m|T, M, E) = \textsc{Constrainer}(\hat{E}, \hat{C}_m, \hat{S}_m, \hat{F}_m)$$

where C_m, S_m, F_m are the lists of predicted countries, states, and feature types for m, and ATTRIBUTEPREDICTOR, CANDIDATEGENERATOR, and CONSTRAINER are defined in the following sections.

6.2.1 Attribute predictor

This function predicts the country, state, and feature type of m using a model that combines prompting and a masked language modeling objective. The prediction targets are defined as:

Feature Class is one of the nine types defined by GeoNames: A, Administrative boundaries (e.g., countries, states, provinces); P, Populated places (e.g., cities, towns, villages); U, Undersea features (e.g., oceanic ridges, trenches), etc.

State is the canonical name of one of the 3871 first-order administrative divisions in GeoNames, such as states, provinces, or regions.

Country is the canonical name of one of the 252 countries in GeoNames.

We implement prediction of these targets as:

$$Z = \textsc{Transformer}(\textsc{ToInput}(m, M))$$

$$\hat{C}_m = \text{softmax}(Z_c W_c)$$

$$\hat{S}_m = \text{softmax}(Z_s W_s)$$

$$\hat{F}_m = \text{softmax}(Z_f W_f)$$

where ToInput produces text of the form `[CLS] This document discusses these location mentions:` m_1, m_2, ..., $m_{|M|}$ `in which` m `is <mask> located in <mask> of <mask> [SEP]`, f, s, c, are the indexes of the three `<mask>` tokens, $W_c, W_s, W_f \in \mathbb{R}^{N \times H}$ are the learnable parameters of the three classification heads, N is the size of the transformer tokenizer's vocabulary, and H is the size of the transformer's contextualized representations. We add new tokens to the transformer's tokenizer to ensure every country, state, and feature type is a single token, e.g., *United States*.

The model is trained with the classification loss:

$$L = C_m log(\hat{C}_m) + S_m log(\hat{S}_m) + F_m log(\hat{F}_m)$$

where C_m, S_m, and F_m are one-hot vectors of size $|N|$ representing the true country, state, and feature type for mention m. At prediction time, we constrain the outputs of the softmax to the subset of the vocabulary appropriate for each prediction type. For example, when the model predicts the word for the country `<mask>`, only the 252 country names are allowed to be non-zero.

We train this model on the labeled data in the toponym datasets. Optionally, we also pre-train (before the fine-tuning) on additional data that we synthesize directly from the GeoNames ontology following the prompt format of ToInput. See section 6.3.1 for details.

Algorithm 3: Constrained Entry Selection

Input: a list of candidate entries, $\hat{E}_m$
top 3 predicted countries, $\hat{C}_m$
top 3 predicted states, $\hat{S}_m$
top 3 predicted feature classes, $\hat{F}_m$
Output: selected candidate entry $\hat{e}$

Def SCORE(x, L)**:**
  if $x = L_0$ **then return** 2
  else return $x \in L$
end
Def ENTRYKEY(e)**:**
  $c \leftarrow$ COUNTRY(e)
  $s \leftarrow$ STATE(e)
  $f \leftarrow$ FEATURE(e)
  $key_1 \leftarrow$ SCORE$(c, \hat{C}_m) \cdot$ SCORE$(s, \hat{S}_m)$
  $key_2 \leftarrow (c \in \hat{C}_m) \cdot (s \in \hat{S}_m) \cdot$ SCORE$(f, \hat{F}_m)$
  return $(key_1, key2)$
end
return MAX$(\hat{E}_m,$ KEY $=$ ENTRYKEY$)$

6.2.2 Candidate generator

We adopt the candidate generator from chapter 3.

6.2.3 Constrainer

Algorithm 3 defines our process for sorting the output of the candidate generator (entries) using the output of the attribute predictor (countries, states, and feature classes). We define the SCORE of a prediction as 2 if it was the top ranked prediction, 1 if it was the second or third ranked prediction, and 0 otherwise. Entries are then sorted by the product of the country and state SCOREs, with the SCORE of the feature class used to break ties. Intuitively, this prefers candidate entries that match both the top predicted

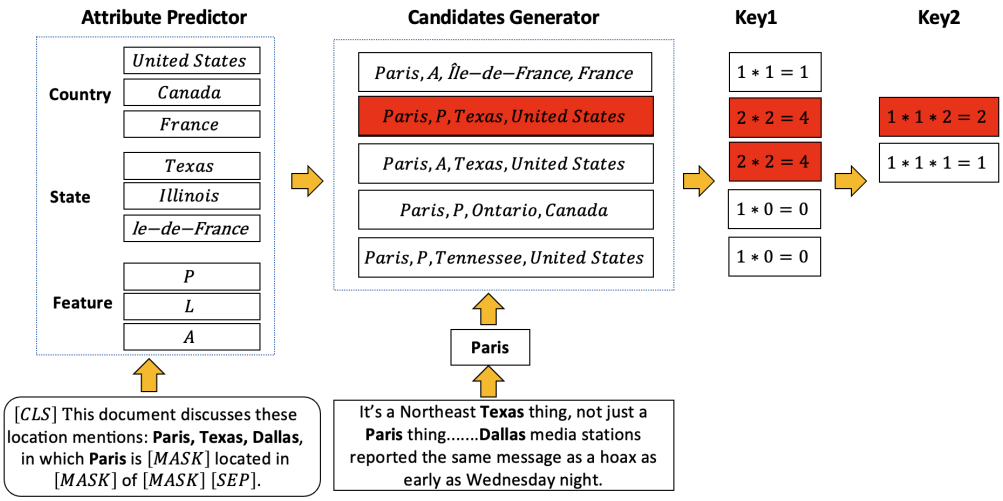

Figure 6.2: An example application of the Constrainer.

country and state, followed by candidate entries that match countries and states from the top 3 predictions. We use a stable sort, so candidates that are assigned the same score retain their population-based sorting from the candidate generator. A worked example of our constrainer algorithm is shown in fig. 6.2. Variants of this algorithm are discussed in section 6.3.2.

6.3 Experiments

The datasets, evaluation metrics and previous geocoding systems has been introduced in chapter 3, chapter 4 and chapter 5.

Before evaluating on the test sets, we performed model selection on the development sets as described in section 6.3.2.

6.3.1 Implementation details

We implement the attribute predictor with the PyTorch[1] v1.7.0 APIs in Huggingface Transformers v2.11.0 (Wolf et al., 2020), using `bert-base`. We train with the AdamW optimizer, a learning rate of 5e-6, a maximum sequence length of 256 tokens, and a number of epochs of 40. When training, we use one NVIDIA A100 GPU with 40G memory and a batch size of 64. The total number of parameters in our model is 112M and the training time is about 0.15 hours.

When synthesizing data from the geographical ontology for pre-training, we filtered all cities, states and countries with less than 100 population and take the entries from some other special feature classes, such as H (stream, lake), L (parks, area) and T (mountain, hill, rock). To construct the input for pre-training, we used the same prompt as for finetuning and sampled a different number of locations (from 2 to 15) within the same country as the document mentions. Most of the hyperparameters are same with finetuning just the batch size is 32 and training epochs is 10.

When using a generative sequence-to-sequence objective as described in section 6.4 instead of a masked language modeling objective, we utilize `bart-large` with the PyTorch v2.0.0 APIs in Huggingface Transformers v4.11.3 (Wolf et al., 2020) and FAIRSEQ v0.12.2 (Ott et al., 2019). We train with the AdamW optimizer, a initial learning rate of 1e-5, a learning rate scheduler type of polynomial, a maximum sequence length of 1024 tokens, and the steps of training of 40000. When training,

[1] `https://pytorch.org/`

we use one NVIDIA A100 GPU with 40G memory and a batch size of 8. During evaluation, we use beam search with a beam size of 5. The total number of parameters in our model is 406M and the training time is about 1.3 hours. We use one model to generate only one attribute, when we generate the country name, we use the prompt `[CLS] This document discusses these location mentions:` m_1, m_2, `...,` $m_{|M|}$. `Which country is START` m `END located ?`, the prefix prompt for output generation is m `is located in`. When we generate the state name, we use the prompt `[CLS] This document discusses these location mentions:` m_1, m_2, `...,` $m_{|M|}$. `Which state is START` m `END located ?`, the prefix prompt for output generation is m `is located in`. When we generate the feature class, we use the prompt `[CLS] This document discusses these location mentions:` m_1, m_2, `...,` $m_{|M|}$. `Which feature class does START` m `END belong to ?`, the prefix prompt for output generation is m `belong to`

6.3.2 Model selection

Model	Accuracy		
	LGL (dev)	GeoWebNews (dev)	TR-News (dev)
GeoPLACE (alg1 top553)	.885	.811	.926
GeoPLACE (alg1 top553 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg2 top553)	.885	.815	.926
GeoPLACE (alg2 top553 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg3 top553)	.900	.815	.926
GeoPLACE (alg3 top553 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg3 top555)	.900	.815	.926
GeoPLACE (alg3 top555 synthesized pre-training)	.902	**.872**	.912
GeoPLACE (alg3 top444)	.893	.815	**.941**
GeoPLACE (alg3 top444 synthesized pre-training)	**.912**	**.872**	.912
GeoPLACE (alg3 top333)	.893	.826	**.941**
GeoPLACE (alg3 top333 synthesized pre-training)	**.912**	.868	.926

Table 6.1: Model selection on the development sets. The top performance on each dataset is in bold, the second best performance is underlined.

For the attribute predictor, we explored a small number of learning rates (1e-6, 2e-6, 5e-6, 1e-5) and number of epochs (10, 20, 30, 40). The best learning rate and number of epochs was selected based on accuracy on the attribute prediction task (not on the full geocoding task).

For the constrainer, we explored three different ways to define key_1 and key_2.

alg3 defines key_1 and key_2 as in algorithm 3.

alg2 allows scores to range from 0 to the length of the list, rather than just from 0 to 2. It defines:

$$key_1 \leftarrow \text{RINDEX}(c, \hat{C}_m) \cdot \text{RINDEX}(s, \hat{S}_m)$$

$$key_2 \leftarrow (c \in \hat{C}_m) \cdot (s \in \hat{S}_m) \cdot \text{RINDEX}(f, \hat{F}_m)$$

$$\textbf{Def } \text{RINDEX}(x, L)\text{: } \textbf{if } x \notin L \textbf{ then } 0$$

$$\textbf{else } |L| - lst.index(val)$$

alg1 prioritizes matching the first country, and also allows scores to range from 0 to the length of the list. It defines:

$$key_1 \leftarrow (c = \hat{C}_{m_0}) \cdot \text{RINDEX}(s, \hat{S}_m)$$

$$key_2 \leftarrow (c \in \hat{C}_m) \cdot (s \in \hat{S}_m) \cdot \text{RINDEX}(f, \hat{F}_m)$$

Table 6.1 shows that there were not large differences between these algorithms in terms of accuracy, but alg3 performed slightly better.

For the constrainer, we also explored four different ways to define the number of predictions to consider in the constrainer.

top3 Only the top 3 countries, states, and feature classes are considered

top4 Only the top 4 countries, states, and feature classes are considered

top5 Only the top 5 countries, states, and feature classes are considered

top553 The top 5 countries, top 5 states, and top 3 feature classes are considered

Table 6.1 shows that there were not large differences between these strategies in terms of accuracy, but top3 performed slightly better.

For the constrainer, we also explored whether or not it helps to pre-train on synthesized data before fine-tuning on the toponym resolution datasets. Table 6.1 shows that pre-training on synthesized data consistently helped on LGL and GeoWebNews but led to small drops in performance on TR-News.

6.4 Results

The top half of table 6.2 compares our model to the existing state-of-the-art on LGL, GeoWebNews, and TR-News. GeoPLACE outperforms prior work by large margins (more than 30% error reduction) on LGL and TR-News, while achieving similar performance on GeoWebNews.

The bottom half of table 6.2 shows an ablation of our model. Pre-training on synthesized data provides small but consistent gains across all datasets. We also try replacing our masked language modeling objective with a sequence-to-sequence style generative objective, asking the model to directly produce `is <feature-type> located in <state> of <country>`. (See section 6.3.1 for prompting details.) This approach yields worse performance than our masked language modeling approach both when fine-tuning BART-large and when using GPT3 in zero-shot mode.

Model	LGL (test)				GeoWebNews (test)				TR-News (test)			
	Acc	A161	Err	AUC	Acc	A161	Err	AUC	Acc	A161	Err	AUC
ReFinED	.576	-	-	-	.658	-	-	-	.720	-	-	-
ReFinED (fine-tuned)	.786	-	-	-	.782	-	-	-	.858	-	-	-
GeoNorm GRPTMCD	.807	.824	46	.135	**.828**	.862	**55**	.114	.918	.933	34	.057
GeoPLACE (ours)	**.863**	**.894**	**21**	**.084**	.822	**.878**	57	**.112**	**.947**	**.957**	**18**	**.038**
GeoPLACE (-synthesized pre-training)	.851	.886	24	.093	.809	.864	63	.123	.904	.922	20	.062
GeoPLACE (-masking, +generative fine-tuned BART)	.633	.696	111	.250	.704	.776	92	.191	.727	.812	95	.167
GeoPLACE (-masking, +generative zero-shot GPT-3)	.733	.795	80	.176	.719	.811	85	.171	.830	.869	63	.115

Table 6.2: Performance on the test sets. Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). We do not report distance-based metrics for ReFinED as this extraction+disambiguation system does not make predictions for all mentions. The best performance on each dataset+metric is in bold.

We release our model for English geocoding under the Apache License v2.0, for off-the-shelf use at `https://github.com/clulab/geonorm`.

6.4.1 EUPEG results

We also report results using the Extensible and Unified Platform for Evaluating Geoparsers (EUPEG; Wang and Hu, 2019). This platform evaluates not geocoders, but geoparsers, where a model must both detect locations and match them to ontology entries. So we couple our geocoder with the best location detection model on EUPEG, the StanfordNER system.

This platform reports several metrics that are incomparable across systems. Accuracy, accuracy@161km, mean error, and area under the error distances curve are all calculated only over locations that were detected, so that a model that detects only 1% of locations but matches 100% of them to their correct ontology entries would get perfect values for these scores, while a model that detects 100% of locations and matches 90% of them to their correct ontology entries would score lower. We nonetheless report these incomparable metrics as EUPEG provides no alternative. EUPEG results are shown in table 6.3

6.5 Qualitative Analysis

Table 6.4 presents a qualitative analysis of errors encountered by our generate-and-rerank system featuring a two-stage document context (GRCD) and our latest state-of-the-art model, GeoPLACE.

	LGL (test)						GeoWebNews (test)					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.776	.353	.486	.775	60	.187	.787	.520	.626	**.944**	33	**.056**
StanfordNER + Pop	.762	.635	.692	.592	135	.360	.866	.648	.741	.673	86	.257
StanfordNER + GeoPLACE	.762	.635	.692	**.888**	**23**	**.109**	.866	.648	.741	.929	**30**	.072
	TR-News (test)						GeoVirus					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.752	.592	.663	.844	78	.121	.860	.559	.678	.807	44	.319
StanfordNER + Pop	.906	.752	.822	.651	119	.287	.927	.903	.915	.655	79	.378
StanfordNER + GeoPLACE	.906	.752	.822	**.967**	**15**	**.033**	.927	.903	.915	**.837**	**23**	**.297**
	WikToR						GeoCorpora					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.230	.298	.259	.591	217	.378	.832	.505	.628	.848	96	.140
StanfordNER + Pop	.209	.540	.301	.184	460	.702	.899	.526	.664	.676	106	.270
StanfordNER + GeoPLACE	.209	.540	.301	**.629**	**171**	**.342**	.899	.526	.664	**.875**	**48**	**.122**
	Hu2014						Ju2016					
Model	Pre	Rec	F1	A161	Err	AUC	Pre	Rec	F1	A161	Err	AUC
Edinburgh	.486	.656	.559	**.114**	**86**	**.607**	.000	.000	.000	—	–	—
StanfordNER + Pop	.504	.788	.615	.000	228	.758	.162	.010	.019	0.0	**203**	**.74**3
StanfordNER + GeoPLACE	.504	.788	.615	.071	92	.632	.162	.010	.019	**.046**	354	.768

Table 6.3: Performance on the test sets. Precision (Pre), Recall (Rec), and F1 are on the location detection task, while the other metrics are on the geocoding task Higher is better for accuracy (Acc) and accuracy@161km (A161). Lower is better for mean error (Err) and area under the error distances curve (AUC). The best performance on each dataset and geocoding metric is in bold.

The first row displays an example where GRCD falls short while GeoNorm excels. This can be attributed to GeoNorm's superior ability to employ masked language models for accurately predicting the countries, states, and feature codes of toponyms in the text prior to their resolution.

The second row portrays an instance where our most proficient model, GeoPLACE, experiences a failure. This occurs because predicting feature codes with the aid of a

masked language model proves to be more challenging compared to predicting countries and states. Thoroughly resolving this problem is likely to necessitate improvements in the prediction performance for all types of geographical metadata.

6.6 Conclusion

We introduced a new paradigm for geocoding where instead of trying to map directly from text to an ontology entry, we predict geographical attributes and use those to deterministically constrain the set of valid ontology entries. Our approach leads to large error reduction over the current state-of-the-art on the LGL and TR-News datasets.

	Example	Candidate					Rank	
		Name	Pop.	Type	State	Country	GRCD	GeoPLACE
1	*But the Mt. Pleasant News has reviewed legal documents.....one from Winfield police chief John Chaney,.....he writes, as do my efforts to insure <u>New London</u> is a safe community.*	New London County	274055	ADM2	Connecticut	United States	1	2
		New London	27179	PPL	Connecticut	United States	2	3
		New London	7172	PPL	Wisconsin	United States	3	4
		New London	1882	PPL	Iowa	United States	4	1
2	*John-Paul Delaney (18) of Upper Abbey Street, Cahir, is charged with assault, assault causing harm and theft of a mobile phone at Main Street, <u>Tipperary</u>, on the same date.*	Tipperary	159553	ADM2	Munster	Ireland		1
		Tipperary	4979	PPL	Munster	Ireland		2
		Tipperary	0	HMSD	Western Australia	Australia		3
		Tipperary	0	HMSD	New South Wales	Australia		4

Table 6.4: Examples of predictions from our generate-and-rerank system with 2-stage document context (GRCD) and our new SOTA model, GeoPLACE. Target location mentions are underlined. Human annotated ontology entries are in bold. (ADM2 represents a county, PPL represents a city, HMSD represents a residence specific to Australia and New Zealand)

CHAPTER 7

Future Work

In this thesis, we have explored the potential of using the better candidate generation, transformer-based reranking, two-stage resolution and a new paradigm for improving toponym resolution. We have made significant advancements in the field, and our proposed methods have shown promising results. However, there is still a vast scope for improvement and exploration in this domain. In this chapter, we outline potential future work that can build upon the foundation laid by this research. Specifically, we focus on three main areas: (1) building an end-to-end system for toponym recognition and normalization, (2) adapting large language models for toponym resolution, and (3) generalizing our approach to other domains.

7.1 Building an End-to-End System for Toponym Recognition and Normalization

Current methods for toponym resolution often involve separate processes for toponym recognition and normalization. One possible direction for future work is to develop an end-to-end system that can perform both tasks simultaneously. This could lead to more accurate and efficient toponym resolution by taking advantage of the interdependencies between the recognition and normalization steps.

To achieve this, researchers could explore various architectures that can effectively

combine the strengths of the individual models for recognition and normalization. For example, an attention-based mechanism can be employed to allow the model to focus on specific input features relevant to each task. Additionally, multi-task learning strategies can be explored to enable the model to learn shared representations that benefit both tasks.

7.2 Adapting Large Language Models for Toponym Resolution

Large language models like GPT-4 have demonstrated remarkable performance in a wide range of natural language processing tasks. Future work could investigate how these models can be adapted or fine-tuned for toponym resolution tasks. This could involve training the large language models on geospatially-annotated datasets to learn contextual information and patterns specific to toponyms.

A potential challenge in this approach is the need for large-scale, high-quality training data. Researchers can address this issue by leveraging existing geospatial databases and resources, as well as by developing novel techniques for automatically generating geospatial annotations in textual data.

7.3 Generalizing the Approach to Other Domains

While our work focuses on toponym resolution, the methods developed in this thesis may also be applicable to other domains that require entity recognition and disambiguation. Future work could explore the potential of our approach for resolving other types of named entities, such as medical concepts, person names, organization names, or temporal

expressions.

To generalize the approach to other domains, researchers can investigate the transferability of the learned features and representations, as well as adapt the model architectures to suit the specific characteristics of the target domain. Moreover, evaluation metrics and benchmarks can be developed to assess the performance of the generalized models across different domains.

7.4 Conclusion

In conclusion, the future work outlined in this chapter has the potential to significantly advance the state-of-the-art in toponym resolution and related fields. By building on the insights and methods developed in this thesis, researchers can continue to explore novel approaches and techniques to improve the accuracy and applicability of neural network-based toponym resolution systems.

APPENDIX A

Appendix

A.1 Appendix for Compositional GeoParsing

The main text of the paper focused on parsing the toponyms to the point-level result, such as coordinates or geographical database entry ID. This appendix contains the work we have done for compositional geoparsing.

Figure A1: The red-shaded area is the polygon label for *Biancavilla*, which is defined by the set of its boundary coordinates retrieved from OpenStreetMap.

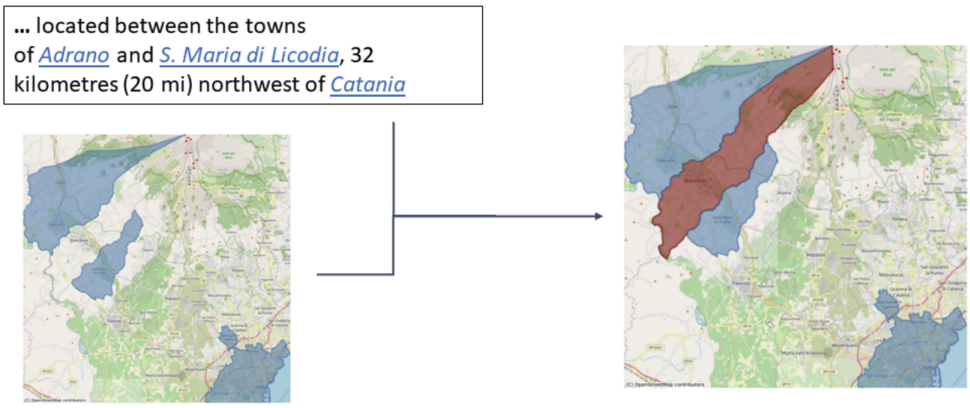

Figure 1: An illustrative example of complex geographical description parsing. Given a text describing a target geolocation the goal is to aproximate its geometry using as reference the geometries of the geolocations that appear in the description (*Adrano*, *S.Maria di Licodia* and *Catania*). In this example, the target geolocation is **Biancavilla**, but in general, there may not be a name for the target geolocation.

Figure A2: (Laparra and Bethard, 2020)

A.1.1 Task Definition

Most existing datasets and systems treat geocoding as a problem of identifying points rather than polygons. Yet the vast majority of real places in geospatial databases are complex polygons (as in fig. A1), not simple points. The GeoCoDe polygon-based dataset (Laparra and Bethard, 2020) includes complex descriptions of locations (e.g., *between the towns of Adrano and S. Maria di Licodia*) and not just explicit place names (e.g., *Paris*). In this task, the goal is to predict the polygon of the target location by using the description and the polygons of reference locations from the description, shown in fig. A2. The current state-of-the-art for complex geographical description geocoding is rule-based.

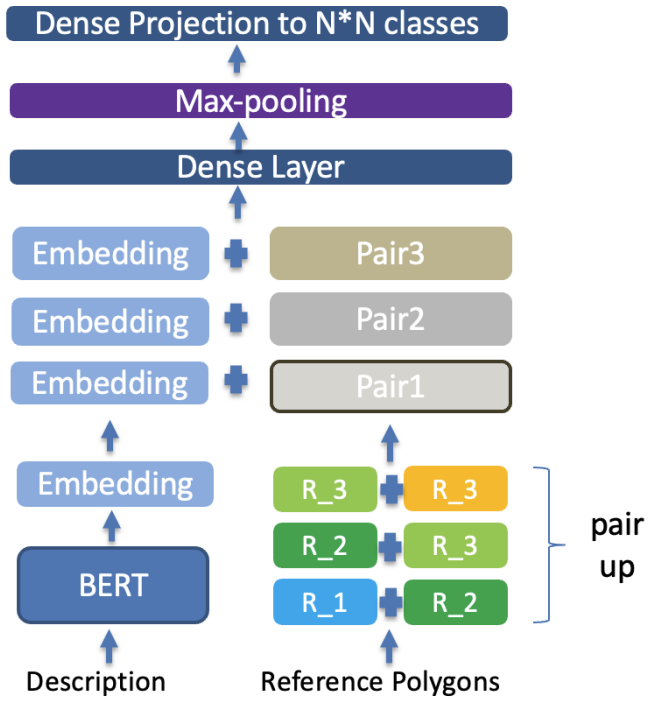

Figure A3: The architecture for World Discretization.

A.1.2 Proposed Methods

We have tried many methods on this compositional geoparsing task, introduced as follows:

World Discretization

- Discretize the whole world in to N*N tiles
- Discretize the relevant area for each case (minimum square area that covers all the reference entities in one description). For example, if the target entity was Tucson, instead of dividing the whole world in N tiles we could just divide Arizona in N tiles. Or if the target entity was Paris, we could divide France in N tiles.

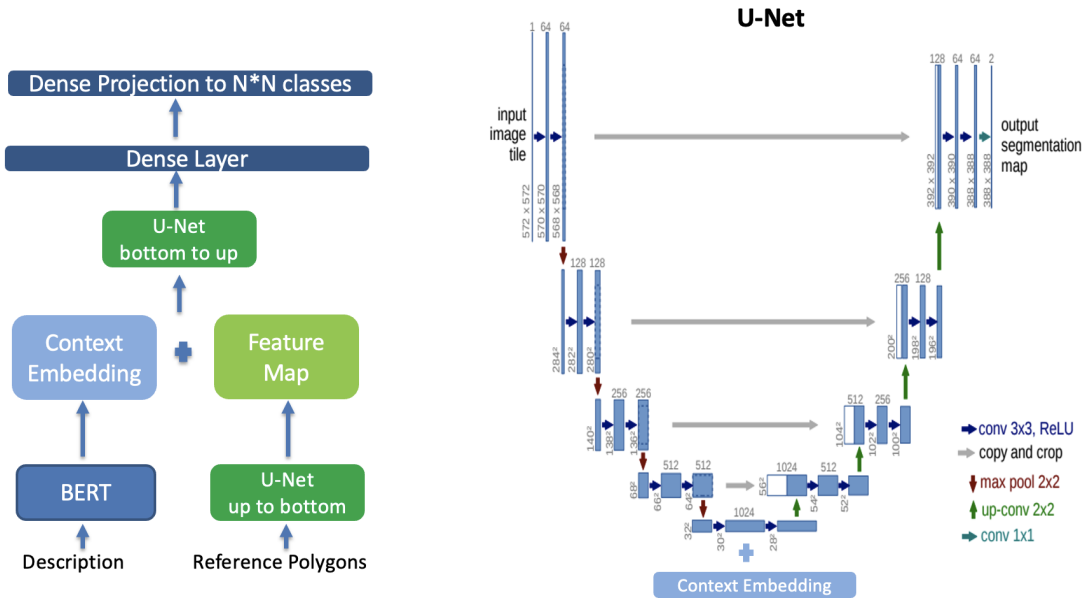

Figure A4: The architecture for World Bitmap.

- Discretize the relevant area for each case by using pair reference locations. Instead of putting the polygons for all of reference locations in one matrix, we pair them up and concatenate their representation and them put them in one matrix.
- For all of the above ideas, we have tried to use 10, 26, 50, 100 as hyperparameter N, for each idea, we have tried to use LOCATION token to be instead of the exact location name.
- The architecture is shown as fig. A3

World Bitmap

- Convert the whole world into one bitmap with N*N size, similar to a digital image. For example, The geometries of one entity are encoded into a N*N bitmap where 1.0 values indicate where the geometry is. For the prediction, instead of a softmax

we should use N*N sigmoids

- Convert the relevant area into one bitmap with N*N size (minimum square area that covers all the reference entities in one description)
- Utilizing U-Net model (shown in fig. A4): U-Net model is one SOTA model for image segmentation, our bitmap idea is very similar to an image segmentation task. We add the BERT representation for description at the bottom of the U-Net model
- We have tried to use U-Net for original bitmap idea and bitmap with relative boundary idea. We also tried to use pair reference location representation. The hyperparameter N we have used are 10, 32, 64, 70, 100, 128, 200, 256. For each idea, we have tried to use LOCATION to be instead of the exact location name.
- The architecture is shown as fig. A4

Relate, Azimuth, Distance and Target Size

- Using the library to convert the geometry of each reference location in description into 3 kinds of classification label: Relate, Azimuth and Distance.
- Predicting the target size for target entity by using description
- Converting the above 4 kinds of information to geometry

However, all of the above methods achieved performance lower than the performance of rule-based methods from Laparra and Bethard (2020). We will explore this topic in the future.

References

Adams, Benjamin and Grant McKenzie (2018). "Crowdsourcing the character of a place: Character-level convolutional networks for multilingual geographic text classification". In: *Transactions in GIS* 22.2, pp. 394–408.

Aldana-Bobadilla, Edwin et al. (2020). "Adaptive Geoparsing Method for Toponym Recognition and Resolution in Unstructured Text". In: *Remote Sensing* 12.18. ISSN: 2072-4292. DOI: 10.3390/rs12183041. URL: https://www.mdpi.com/2072-4292/12/18/3041.

Ardanuy, Mariona Coll and Caroline Sporleder (2017). "Toponym disambiguation in historical documents using semantic and geographic features". In: *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, pp. 175–180.

Ardanuy, Mariona Coll et al. (2019). "Resolving Places, Past and Present: Toponym Resolution in Historical British Newspapers Using Multiple Resources". In: *Proceedings of the 13th Workshop on Geographic Information Retrieval*. GIR '19. Lyon, France: Association for Computing Machinery. ISBN: 9781450372602. DOI: 10.1145/3371140.3371143. URL: https://doi.org/10.1145/3371140.3371143.

Ardanuy, Mariona Coll et al. (2020a). "A Deep Learning Approach to Geographical Candidate Selection through Toponym Matching". In: *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '20. Seattle, WA, USA: Association for Computing Machinery, 385–388. ISBN: 9781450380195. DOI: 10.1145/3397536.3422236. URL: https://doi.org/10.1145/3397536.3422236.

— (2020b). "A deep learning approach to geographical candidate selection through toponym matching". In: *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pp. 385–388.

Ardanuy, Mariona Coll et al. (2022). "A dataset for toponym resolution in nineteenth-century english newspapers". In: *Journal of Open Humanities Data* 8.

Ashktorab, Zahra et al. (2014). "Tweedr: Mining twitter to inform disaster response." In: *ISCRAM*, pp. 269–272.

Ayoola, Tom, Joseph Fisher, and Andrea Pierleoni (July 2022). "Improving Entity Disambiguation by Reasoning over a Knowledge Base". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 2899–2912. DOI: 10.18653/v1/2022.naacl-main.210. URL: https://aclanthology.org/2022.naacl-main.210.

Berico Technologies (2012). *Cartographic Location and Vicinity Indexer (CLAVIN)*. URL: https://clavin.bericotechnologies.com/.

Bhargava, Preeti, Nemanja Spasojevic, and Guoning Hu (Sept. 2017). "Lithium NLP: A System for Rich Information Extraction from Noisy User Generated Text on Social Media". In: *Proceedings of the 3rd Workshop on Noisy User-generated Text*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 131–139. DOI: 10.18653/v1/W17-4417. URL: https://aclanthology.org/W17-4417.

Bruijn, Jens A de et al. (2018). "TAGGS: grouping tweets to improve global geoparsing for disaster response". In: *Journal of Geovisualization and Spatial Analysis* 2.1, p. 2.

Cardoso, Ana Bárbara, Bruno Martins, and Jacinto Estima (2019). "Using Recurrent Neural Networks for Toponym Resolution in Text". In: *EPIA Conference on Artificial Intelligence*. Springer, pp. 769–780.

Cardoso, Ana Bárbara, Bruno Martins, and Jacinto Estima (2022). "A Novel Deep Learning Approach Using Contextual Embeddings for Toponym Resolution". In: *ISPRS International Journal of Geo-Information* 11.1. ISSN: 2220-9964. DOI: 10.3390/ijgi11010028. URL: https://www.mdpi.com/2220-9964/11/1/28.

Cheng, Zhiyuan, 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 on Information and knowledge management*, pp. 759–768.

DeLozier, Grant, Jason Baldridge, and Loretta London (2015). "Gazetteer-Independent Toponym Resolution Using Geographic Word Profiles". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. Austin, Texas: AAAI Press, 2382–2388. ISBN: 0262511290.

DeLozier, Grant et al. (Aug. 2016). "Creating a Novel Geolocation Corpus from Historical Texts". In: *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*. Berlin, Germany: Association for Computational Linguistics, pp. 188–198. DOI: 10.18653/v1/W16-1721. URL: https://aclanthology.org/W16-1721.

Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Fize, Jacques, Ludovic Moncla, and Bruno Martins (2021). "Deep learning for toponym resolution: Geocoding based on pairs of toponyms". In: *ISPRS International Journal of Geo-Information* 10.12, p. 818.

Freire, Nuno et al. (2011). "A metadata geoparsing system for place name recognition and resolution in metadata records". In: *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pp. 339–348.

Gey, Fredric et al. (2005). "GeoCLEF: the CLEF 2005 cross-language geographic information retrieval track overview". In: *Workshop of the cross-language evaluation forum for european languages*. Springer, pp. 908–919.

Gorski, Krzysztof M et al. (2005). "HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere". In: *The Astrophysical Journal* 622.2, p. 759.

Gritta, Milan, Mohammad Taher Pilehvar, and Nigel Collier (July 2018). "Which Melbourne? Augmenting Geocoding with Maps". In: *Proceedings of the 56th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, pp. 1285–1296. DOI: 10.18653/v1/P18-1119. URL: https://aclanthology.org/P18-1119.

Gritta, Milan, Mohammad Taher Pilehvar, and Nigel Collier (2019). "A pragmatic guide to geoparsing evaluation". In: *Language Resources and Evaluation*, pp. 1–30.

Gritta, Milan et al. (June 2017). "What's missing in geographical parsing?" en. In: *Language Resources and Evaluation* 52.2, pp. 603–623. ISSN: 1574-0218. DOI: 10.1007/s10579-017-9385-8. URL: https://doi.org/10.1007/s10579-017-9385-8 (visited on 07/12/2021).

Grover, Claire et al. (2010). "Use of the Edinburgh geoparser for georeferencing digitized historical collections". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1925, pp. 3875–3889.

Halterman, Andrew (2017). "Mordecai: Full Text Geoparsing and Event Geocoding". In: *The Journal of Open Source Software* 2.9. DOI: 10.21105/joss.00091.

Hay, Simon I et al. (2013). "Global mapping of infectious disease". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 368.1614, p. 20120250.

Hoang, Thi Bich Ngoc and Josiane Mothe (2018). "Location extraction from tweets". In: *Information Processing & Management* 54.2, pp. 129–144.

Hosseini, Kasra, Federico Nanni, and Mariona Coll Ardanuy (Oct. 2020). "DeezyMatch: A Flexible Deep Learning Approach to Fuzzy String Matching". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 62–69. DOI: 10.18653/v1/2020.emnlp-demos.9. URL: https://aclanthology.org/2020.emnlp-demos.9.

Hu, Xuke et al. (2023). "How can voting mechanisms improve the robustness and generalizability of toponym disambiguation?" In: *International Journal of Applied Earth Observation and Geoinformation* 117, p. 103191.

Ji, Zongcheng, Qiang Wei, and Hua Xu (2020). “Bert-based ranking for biomedical entity normalization”. In: *AMIA Summits on Translational Science Proceedings* 2020, p. 269.

Jurgens, David et al. (2015). “Geolocation prediction in twitter using social networks: A critical analysis and review of current practice”. In: *Ninth international AAAI conference on web and social media*.

Kamalloo, Ehsan and Davood Rafiei (2018). “A coherent unsupervised model for toponym resolution”. In: *Proceedings of the 2018 World Wide Web Conference*, pp. 1287–1296.

Karimzadeh, Morteza et al. (2013). “GeoTxt: a web API to leverage place references in text”. In: *Proceedings of the 7th workshop on geographic information retrieval*, pp. 72–73.

Katz, Philipp and Alexander Schill (2013). “To learn or to rule: two approaches for extracting geographical information from unstructured text”. In: *Data Mining and Analytics 2013 (AusDM’13)* 117.

Kulkarni, Sayali et al. (2020). “Spatial Language Representation with Multi-Level Geocoding”. In: *arXiv preprint arXiv:2008.09236*.

— (Aug. 2021). “Multi-Level Gazetteer-Free Geocoding”. In: *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*. Online: Association for Computational Linguistics, pp. 79–88. DOI: 10.18653/v1/2021.splurobonlp-1.9. URL: https://aclanthology.org/2021.splurobonlp-1.9.

Kumar, Abhinav and Jyoti Prakash Singh (2019). “Location reference identification from tweets during emergencies: A deep learning approach”. In: *International journal of disaster risk reduction* 33, pp. 365–375.

Laparra, Egoitz and Steven Bethard (Dec. 2020). “A Dataset and Evaluation Framework for Complex Geographical Description Parsing”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 936–948. DOI: 10.18653/v1/2020.

coling-main.81. URL: https://aclanthology.org/2020.coling-main.81.

Lee, Sunshin et al. (2015). "Read between the lines: A Machine Learning Approach for Disambiguating the Geo-location of Tweets". In: *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 273–274.

Leidner, JL (2007). "Toponym resolution: A comparison and taxonomy of heuristics and methods". PhD thesis. PhD Thesis, University of Edinburgh.

Leidner, Jochen L (2021). "A Survey of Textual Data & Geospatial Technology". In: *Handbook of Big Geospatial Data*. Springer, pp. 429–457.

Lieberman, Michael D and Hanan Samet (2011). "Multifaceted toponym recognition for streaming news". In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 843–852.

— (2012). "Adaptive context features for toponym resolution in streaming news". In: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 731–740.

Lieberman, Michael D, Hanan Samet, and Jagan Sankaranarayanan (2010). "Geotagging with local lexicons to build indexes for textually-specified spatial data". In: *2010 IEEE 26th international conference on data engineering (ICDE 2010)*. IEEE, pp. 201–212.

Liu, Fangyu et al. (June 2021). "Self-Alignment Pretraining for Biomedical Entity Representations". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4228–4238.

Luo, Xiangyang et al. (2020). "An overview of microblog user geolocation methods". In: *Information Processing & Management* 57.6, p. 102375.

Mani, Inderjeet et al. (2010). "SpatialML: annotation scheme, resources, and evaluation". In: *Language Resources and Evaluation* 44.3, pp. 263–280.

Martins, Bruno, Ivo Anastácio, and Pável Calado (2010). "A machine learning approach for resolving place references in text". In: *Geospatial thinking*. Springer, pp. 221–236.

Melo, Fernando and Bruno Martins (2017). "Automated geocoding of textual documents: A survey of current approaches". In: *Transactions in GIS* 21.1, pp. 3–38.

Mikolov, Tomás et al. (2013). "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1301.3781.

Monteiro, Bruno R, Clodoveu A Davis Jr, and Fred Fonseca (2016). "A survey on the geographic scope of textual documents". In: *Computers & Geosciences* 96, pp. 23–34.

Ott, Myle et al. (2019). "fairseq: A Fast, Extensible Toolkit for Sequence Modeling". In: *Proceedings of NAACL-HLT 2019: Demonstrations*.

Peters, Matthew E. et al. (June 2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202.

Rayson, Paul et al. (2017). "A deeply annotated testbed for geographical text analysis: The corpus of lake district writing". In: *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*, pp. 9–15.

Santos, João, Ivo Anastácio, and Bruno Martins (2015). "Using machine learning methods for disambiguating place references in textual documents". In: *GeoJournal* 80.3, pp. 375–392.

Santos, Rui et al. (2018). "Toponym matching through deep neural networks". In: *International Journal of Geographical Information Science* 32.2, pp. 324–348.

Smith, David A and Gregory Crane (2001). "Disambiguating geographic names in a historical digital library". In: *International Conference on Theory and Practice of Digital Libraries*. Springer, pp. 127–136.

Speriosu, Michael and Jason Baldridge (2013a). “Text-driven toponym resolution using indirect supervision”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1466–1476.

— (Aug. 2013b). “Text-Driven Toponym Resolution using Indirect Supervision”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1466–1476. URL: https://aclanthology.org/P13-1144.

Tateosian, Laura et al. (2017). “Tracking 19th century late blight from archival documents using text analytics and geoparsing”. In: *Free and open source software for geospatial (FOSS4G) conference proceedings*. Vol. 17, p. 17.

Tobin, Richard et al. (2010). “Evaluation of georeferencing”. In: *proceedings of the 6th workshop on geographic information retrieval*, pp. 1–8.

Wallgrün, Jan Oliver et al. (2018). “GeoCorpora: building a corpus to test and train microblog geoparsers”. In: *International Journal of Geographical Information Science* 32.1, pp. 1–29.

Wang, Jimin and Yingjie Hu (2019). “Enhancing spatial and textual analysis with EUPEG: An extensible and unified platform for evaluating geoparsers”. In: *Transactions in GIS* 23.6, pp. 1393–1419. DOI: https://doi.org/10.1111/tgis.12579. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tgis.12579. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12579.

Wang, Xiaobin et al. (2019a). “Dm_nlp at semeval-2018 task 12: A pipeline system for toponym resolution”. In: *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 917–923.

— (June 2019b). “DM_NLP at SemEval-2018 Task 12: A Pipeline System for Toponym Resolution”. In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, pp. 917–923. DOI: 10.18653/v1/S19-2156. URL: https://aclanthology.org/S19-2156.

Weissenbacher, Davy et al. (June 2019). "SemEval-2019 Task 12: Toponym Resolution in Scientific Papers". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, pp. 907–916. DOI: 10.18653/v1/S19-2155. URL: https://aclanthology.org/S19-2155.

Wolf, Thomas et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

Xu, Dongfang, Zeyu Zhang, and Steven Bethard (July 2020). "A Generate-and-Rank Framework with Semantic Type Regularization for Biomedical Concept Normalization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8452–8464. DOI: 10.18653/v1/2020.acl-main.748. URL: https://aclanthology.org/2020.acl-main.748.

Yan, Zheren et al. (2021). "The Integration of Linguistic and Geospatial Features Using Global Context Embedding for Automated Text Geocoding". In: *ISPRS International Journal of Geo-Information* 10.9, p. 572.

Zhang, Wei and Judith Gelernter (2014). "Geocoding location expressions in Twitter messages: A preference learning method". In: *Journal of Spatial Information Science* 2014.9, pp. 37–70.

Zhang, Zeyu and Steven Bethard (July 2023). "Improving Toponym Resolution with Better Candidate Generation, Transformer-based Reranking, and Two-Stage Resolution". In: *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*. Ed. by Alexis Palmer and Jose Camacho-collados. Toronto, Canada: Association for Computational Linguistics, pp. 48–60. DOI: 10.18653/v1/2023.starsem-1.6. URL: https://aclanthology.org/2023.starsem-1.6.