DHBW
Duale Hochschule
Baden-Württemberg
CAS
Center for Advanced Studies

Global Geoparsing at Street Level
with Dynamically Generated Gazetteers

Thesis

submitted for the degree of
Master of Science (M.Sc.)

in
Computer Science

to the Center for Advanced Studies
of the Baden-Württemberg Cooperative State University

Author:	Ernesto Elsäßer
Matriculation Number:	2016242
Processing Period:	September 2019 - January 2020
Cooperative Partner:	moovel Group GmbH, Stuttgart
1st Supervisor:	Prof. Kay Margarethe Berkling, PhD
2nd Supervisor:	Raphael Reimann, M.A.

Notice of receipt DHBW CAS

Declaration

I hereby assure that this assignment is my own work and that I have not sought or used inadmissible help of third parties to produce this work and that I have clearly referenced all sources used in the work. This work has not yet been submitted to another examination institution neither in the same nor in a similar way and has not yet been published.

I further assure that I uploaded an unencrypted, digital copy of the document (in one of the formats .doc, .docx, .odt, .pdf or .rtf) to Moodle. All digitally submitted copies match the printed version in both content and wording, without exceptions.

Place, Date

Signature

Abstract

This thesis presents a new geoparsing system that specializes in extracting local references. Most gepoarsers available today were designed to detect references to large geographic entities like countries and cities. For locations *inside* cities, those systems generally perform poorly. This thesis first shows why street-level geoparsing requires special techniques and then proposes a new method to recognize and resolve street-level location names. The key innovation of this method is to dynamically infer the geographic scope of the input document. This information is then used to construct local gazetteers that match the geographic vocabulary of the documents target audience. These gazetteers significantly increase recognition and resolution performance for smaller entities in comparison to other state-of-the-art geoparsing systems.

Contents

List of Figures

List of Tables

Acronyms

NLP Natural Language Processing

POS Part of Speech

NER Named Entity Recognition

EL Entity Linking

IE Information Extraction

GIS Geographic Information System(s)

GIR Geographic Information Retrieval

GIE Geographic Information Extraction

AI Artificial Intelligence

ML Machine Learning

HMM Hidden Markov Model

MEMM Maximum Entropy Markov Model

CRF Conditional Random Field

SVM Support Vector Model

CNN Convolutional Neural Network

RNN Recurrent Neural Network

LSTM Long Short Term Memory

Bi-LSTM Bidirectional Long Short Term Memory

Chapter 1

Introduction

Everything happens somewhere. If the things that happen are important, people write about them. If important things happen at places that matter to us, we might want to read about them. But how do we find all texts written about a specific location, without reading everything else as well? The modern approach is to use technology. But software systems can only filter by location if they can link a piece of text to geographic entities. This thesis explores means to create such links.

More technically, this thesis is about geoparsing, the automated extraction of coordinate data from unstructured text. Geoparsing is an active field of research and has enabled a wide range of applications. Applied to tweets, geoparsing can help rescue operations during a disaster [Middleton et al., 2013]. Applied to scientific publications, geoparsing can be used to trace virus migration [Weissenbacher et al., 2015]. Applied to newspaper articles, geoparsing can be used to populate a thematic world map [Adams et al., 2015]. The goal of this thesis is to facilitate a new application: content discovery through location references in media and literature.

1.1 Motivation

The motive of this thesis is exploration - both that of new stories and new places. The idea is that one can inspire the other. Great stories draw people to the places they mention. And great places draw people to the stories that are told about them. Not only fictional stories, but all kinds of stories, from news articles to restaurant reviews. By offering people stories about the places around them, technology can encourage them to explore their environment - both physically and intellectually.

To facilitate such bi-faceted exploration, a software system had to retrieve textual content for a selected location and guide users to mentioned locations in selected texts. This

The rapper Jay-Z loves his hometown New York. He grew up in Brooklyn, but now lives in Tribeca and can be seen driving expensive cars down Broadway and 8th Street. His old apartment was located on the corner of State Street and Flatbush Avenue in Bed-Stuy. In his song "Empire State of Mind" he mentions the Statue of Liberty, the Empire State Building and the World Trade Center.

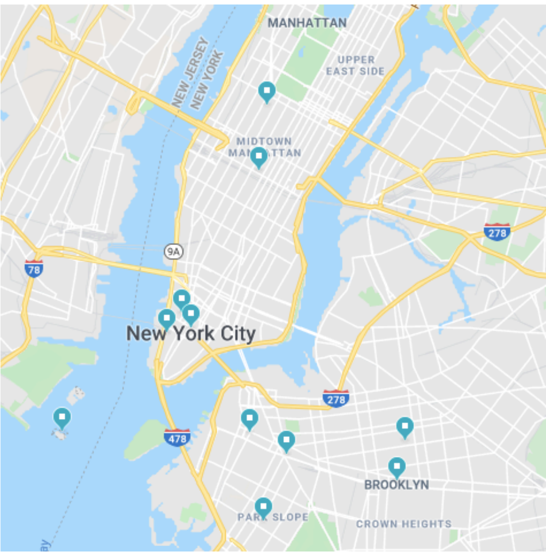


Figure 1.1: Sample geoparsing task

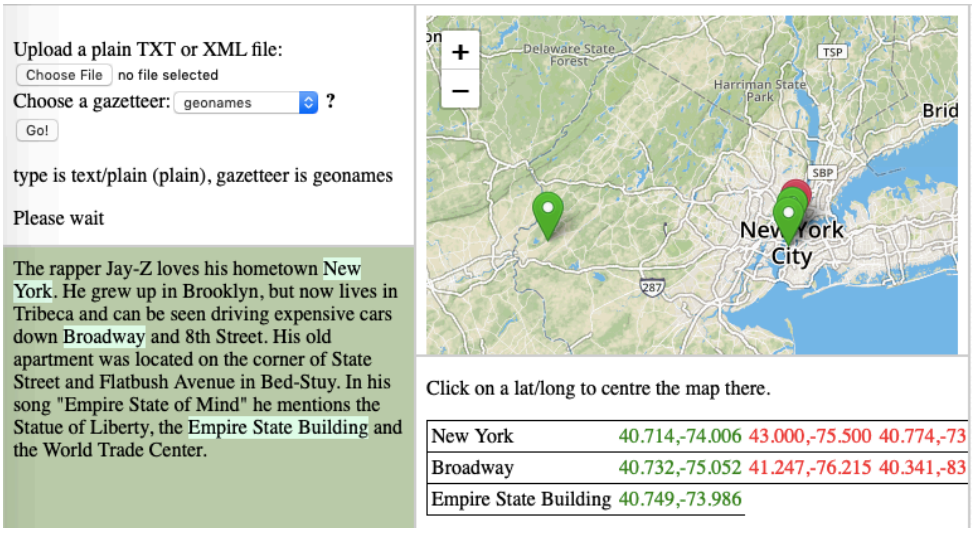
requires the association of location references in textual form with the coordinates these references represent. Figure 1.1 illustrates this task. The sample text on the left side mentions several locations in New York City. The map on the right side shows the coordinates of the mentioned locations. Automatically deriving the information on the right from the information at left, at its core, a geoparsing problem.

The map in Figure 1.1 was prepared manually. For a person familiar with the area, this task is trivial. For state-of-the-art geoparsers, however, this short sample text is quite challenging. As an example, Figure 1.2 shows the results produced by the well-known Edinburgh Geoparser[1]. While modern systems admittedly achieve better results, it will be shown (in Chapter 4) that no currently available geoparsing system is capable of correctly resolving all references in this short sample text. The difficult names are not the ones that refer to large, populous entities like cities or city districts, but the ones that refer to small, local entities like buildings or streets. For the scenario of content discovery however, these **street level** entities are essential, because they represent specific spots to visit and discover, as opposed to large areas or regions with no specific focus.

The realization of the envisioned content discovery system therefore requires the development of a geoparser that is capable of extracting street-level references[2]. This is the motivation and goal of this thesis.

[1] Demo available at `http://jekyll.inf.ed.ac.uk/geoparser/`

[2] Meant here is world-wide street-level geoparsing. Some earlier systems do recognize street-level entities within predefined areas [Middleton et al., 2013].



Upload a plain TXT or XML file:

Choose a gazetteer: geonames ?

type is text/plain (plain), gazetteer is geonames

Please wait

The rapper Jay-Z loves his hometown New York. He grew up in Brooklyn, but now lives in Tribeca and can be seen driving expensive cars down Broadway and 8th Street. His old apartment was located on the corner of State Street and Flatbush Avenue in Bed-Stuy. In his song "Empire State of Mind" he mentions the Statue of Liberty, the Empire State Building and the World Trade Center.

Click on a lat/long to centre the map there.

New York	40.714,-74.006	43.000,-75.500	40.774,-73
Broadway	40.732,-75.052	41.247,-76.215	40.341,-83
Empire State Building	40.749,-73.986		

Figure 1.2: Sample geoparsing output (Edinburgh Geoparser, December 2019)

1.2 Problem Statement

The field of Natural Language Processing studies methods to derive structured information from unstructured text. But to access the geographical information embedded in text documents, the techniques of Natural Language Processing alone are not sufficient. Geoparsing systems therefore combine linguistic extraction techniques with ontological knowledge about physical locations. The amount of geographic information that can be extracted by geoparsers depends on the width and depth of the ontological data sources that are used to interpret the location names obtained from linguistic analysis.

The more specific the target audience of a text, and the more granular the referenced locations, the more local knowledge must a system possess to correctly extract all mentions. The goal of this thesis is to find ways for geoparsing systems to recognize and resolve highly local location references, without limiting the geographic scope of the system or using a priori knowledge about a document's origin.

1.3 Scope

This thesis investigates how geoparsers can translate street-level location references in unstructured text into coordinates. As natural language is constantly evolving and infinitely bendable, it should be specified which kinds of texts and which kinds of references this thesis will focus on.

Language allows the use of arbitrary terms to refer to arbitrary locations. The goal of this thesis is to design a system that recognizes official or canonical place names. No attempts will be made to recognize alternative, abbreviated, contracted, colloquialized, paraphrased, compositional, fictional or misspelt names. Official names not in use anymore, i.e. historical names, will not be considered either. The geoparsing techniques used in this thesis further assume grammatically correct texts. Adapting Natural Language Processing tools and geoparsing systems to informal spelling, as for example common in social media, is an active field of research but beyond the scope of this thesis. For the use case described above, the most relevant text formats are published articles and works of literature, which tend to follow customary rules of form and style.

It must further be defined what this thesis considers to be *street-level* entities. As will be shown in Chapter 4, existing geoparsing systems already perform well for globally significant locations such as continents, countries, administrative districts, cities and other populated places - a group that can be summarized as *geopolitical entities.* A common feature of geopolitical entities is that they have a population. The size of this population - which can range from several billions for continents to only a handful of people for the smallest hamlets and suburbs - defines the category they are placed in. Street-level entities are all locations that fall below this scheme of classification because they do not have a population in the political sense (they might have inhabitants). Described more graphically, street-level entities are the spots and shapes that become visible when a digital map is zoomed into a populated area. The term “street-level” was chosen because most of these places are located along (or are) streets, whereas geopolitical entities contain streets. Examples for street-level entities are all kinds of public and private facilities such as sights, stations, stadiums, malls, parks and office buildings, but also streets, bridges and tunnels.

1.4 Contributions

The main contribution of this thesis is a new geoparsing technique that can extract location references which are too granular or context-specific for most comparable systems. This technique grants automated text processing systems access to a whole new layer of geographical information contained but previously locked away in unstructured text. To make this technology available to the public, the thesis also presents and documents a ready-to-use open-source implementation of the described technique. This technology can be used to develop new applications that connect plain text documents to specific locations within cities and towns. The databases that are used to resolve location references to geographic entities (GeoNames and OpenStreetMap) further provide rich metadata for the referenced locations, allowing applications to inform their users about the locations they read about.

1.5 Document Structure

The remaining chapters of this thesis are organized as follows.

Chapter 2 discusses the theoretical background of geoparsing, covering both Natural Language Processing and Geographic Information Extraction. Chapter 3 surveys previous work that provides solutions, strategies or insights for the construction of new geoparsing systems. It also gives an overview of recent developments in geoparsing and current state-of-the-art systems. Chapter 4 defines which types of location references a street-level geoparser should be able to extract and analyzes whether existing systems can accomplish this task. Chapter 5 then presents a new geoparsing technique capable of extracting such street-level references. The chapter first introduces the theoretical framework of this technique and then describes an integrated software system that implements this technique. Chapter 6 measures the geoparsing performance of that system and compares it to other state-of-the-art systems. Chapter 7 summarizes the results and learnings of the thesis and discussed what remains undone for future work.

Chapter 2

Background

Geoparsing is an interdisciplinary task. It combines techniques from Natural Language Processing and knowledge from Geographic Information Systems. This chapter introduces concepts from both fields that play a role in the design, implementation and evaluation of geoparsers.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a subfield of linguistics, computer science, information engineering and artificial intelligence concerned with processing, analyzing and understanding natural language data. NLP relies on machine learning techniques to label, classify and predict sequences of unstructured text. Since the input of a geoparser is unstructured text, the first processing steps of geoparsing systems almost always involve NLP technology. This section will provide an overview of the most important models, tasks and corpora used in modern NLP applications.

2.1.1 Sequence Labeling

Due to the sequential nature of language, many tasks in the field of NLP can be modeled as sequence labeling problems. This section briefly introduces a range of graphical models frequently used in geoparsing-related NLP tasks.

Sequence labeling is a classification problem. For each element of a sequence, the task is to select the correct label from a predefined label set (e.g. the part of speech). This task is well suited for supervised learning techniques. Using large datasets of labeled text, it is possible to train probabilistic classifiers that learn to predict the correct labels with high accuracy.

An intuitive way to mathematically model a passage of text is a linear graph. Therefore linear graphical models have become a common choice for sequence labeling tasks. An early graphical model that found various applications in NLP is the **Hidden Markov Model** (HMM). Leveraging Bayes' Theorem, HMMs compute the joint probability of element-label pairs from the conditional probabilities of the element given the label and the label given the previous label. For classification, HMMs compute the probabilities of all possible label sequences for the observed input sequence. The most likely set of labels is then selected as prediction.

HMMs have been a popular model for sequence labeling because they are mathematically simple, computationally cheap and can generalize well even from small training sets. They are also undirected, which allows them to process the input sequence bidirectionally (i.e. from start to end and from end to). This technique is often being used to increase the classification performance of already existing models [Sutton et al., 2012]. HMMs do, however, make strong independence assumptions, which can be a problem when working with large feature sets.

HMMs are *generative* classifiers. Such models first estimate prior and conditional probabilities of all possible sequence elements and labels from training data. The prediction algorithm then determines the label sequence that is most likely to generate the observed sequence. In contrast, *discriminative* classifiers directly learn which input features are most useful to discriminate between the possible labels. To train discriminate classifiers it is therefore not necessary to estimate joint probabilities [Sutton et al., 2012]. This has several advantages when working with large sample sizes and feature sets. Consequently, discriminative models have replaced generative models in many areas [Jurafsky and Martin, 2019].

A discriminative graphical model that performs well in many NLP tasks is the Conditional Random Field (CRF) [Lafferty et al., 2001]. Like HMMs, CRFs are probabilistic undirected graphical models that learn probability distributions from labeled training data. A key advantage of CRFs is that they can use large amounts of features. These features can further access every element in the sequence, not just those in a window around the current element. This allows CRFs to take into account a much wider context for the labeling decision [Jurafsky and Martin, 2019].

Figure 2.1, reproduced from Sutton et al. [2012], shows how HMMs and CRFs are related. In the diagram squares represent factor nodes, circles represent variable nodes and shaded circles indicate observed variables. Both models are derived from simpler non-sequential classifiers (Naive Bayes and Logistic Regression) and can be generalized to arbitrary graphical structures. Each column forms a *generative-discriminative pair*.

Although less strong, CRFs still make independence assumptions to allow efficient com-

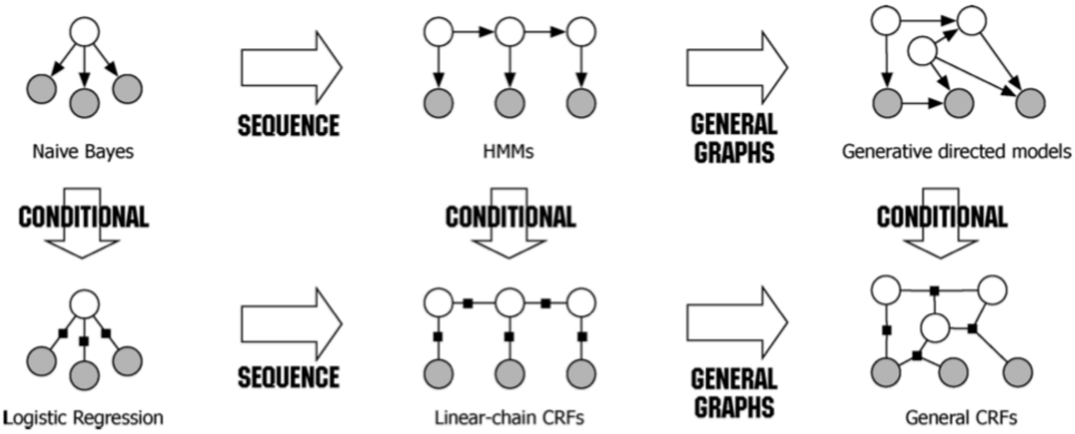


Figure 2.1: Relationships between graphical models for sequence labeling from [Sutton et al., 2012]

putation. When dealing with very large feature sets, this can decrease classification performance. A model that is less sensitive to complex dependencies among features is the Support Vector Machine (SVM). For this reason, SVMs are sometimes preferred over CRFs or HMMs. SVMs are non-probabilistic linear classifiers that construct a hyperplane which optimally separates the instances of two classes in a multidimensional space.

Artificial Neural Networks

In the last decade, another graphical model has become increasingly popular in NLP: artificial neural networks [Yang et al., 2018]. Originally modeled after the synaptic structures in the human brain, neural networks consist of artificial neurons that are arranged in interconnected layers. Each neuron represents a mathematical function that computes a weighted sum of its input signals and propagates it to the next layer. The weights are trained using backpropagation.

Neural networks in which signals are propagated strictly in one direction from the input layer through an arbitrary number of hidden layers to the output layer are called *feedforward* networks. Feedforward networks that use a large number of such hidden layers are often described as *deep* neural networks. For networks that are at the same time deep and wide (i.e. have many neurons per layer), the large number of variable weights can cause overfitting. Convolutional Neural Networks (CNN) are deep neural networks in which the hidden layers are structured into vertically and horizontally contained homogenous groups. The output signals of those groups are only combined at later layers, allowing each group to operate in separation. This approach does not only reduces the overall number of synaptic connections but also allows CNNs to model multiple levels of abstraction, which makes them ideal for pattern recognition. They are therefore very popular in Computer Vision [LeCun et al., 2015], but have also found many applications in NLP [Collobert and Weston, 2008].

The class of neural networks that most naturally model sequence labeling problems are Recurrent Neural Networks (RNN). These networks process an input sequence one element at a time, maintaining an internal state that represents the complete history of previously observed elements [LeCun et al., 2015]. Through this state, RNNs are able to learn dependencies between sequence elements even if they are not directly connected [Boden, 2002]. The distance over which such an internal state can be preserved is, however, limited. Hochreiter [1991] and Bengio et al. [1994] both show that RNNs struggle to learn dependencies over longer intervals because the backpropagated gradients are scaled at each time step, leading to either exploding or vanishing gradients.

The answer to this problem are Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997]. This variant of RNNs introduces an explicit memory unit. This unit acts like an accumulator and can selectively add or remove knowledge in every iteration [LeCun et al., 2015]. Today, many state-of-the-art NLP systems use Bi-LSTM models, which combine two LSTM networks to process the input sequence in both directions [LeCun et al., 2015; Kiperwasser and Goldberg, 2016; Yang et al., 2018; Devlin et al., 2018].

2.1.2 Language Models

Statistical language models are models that assign probabilities to sequences of words. Therefore, they can be used to predict which word is most likely to follow a given input sequence.

N-gram Language Models

N-gram language models follow the assumption that the probability of a given word to follow a sequence of words can be approximated by looking only at the last n words (i.e. bi-gram, tri-gram, etc.). This simplification allows the construction of language models simply by counting all n-grams in a corpus of text. HMMs can be used to both train models and make predictions based on the trained probabilities.

While simple in theory, n-gram models quickly become computationally expensive for higher n values, because the number of parameters increases exponentially with the observed word window [Jurafsky and Martin, 2019]. N-gram models also cannot generalize from training data. Neural language models solve both these problems.

Neural Language Models

Neural language models use neural networks to predict the probability of words to succeed a given input sequence. To allow the network to generalize from the identity of a word to its meaning and role in the sequence, neural language models use so-called word embeddings. Embedding algorithms project words into a continuous, multi-dimensional vector space in which words with similar meanings have similar representations. These vectors then serve as input features for the neural network.

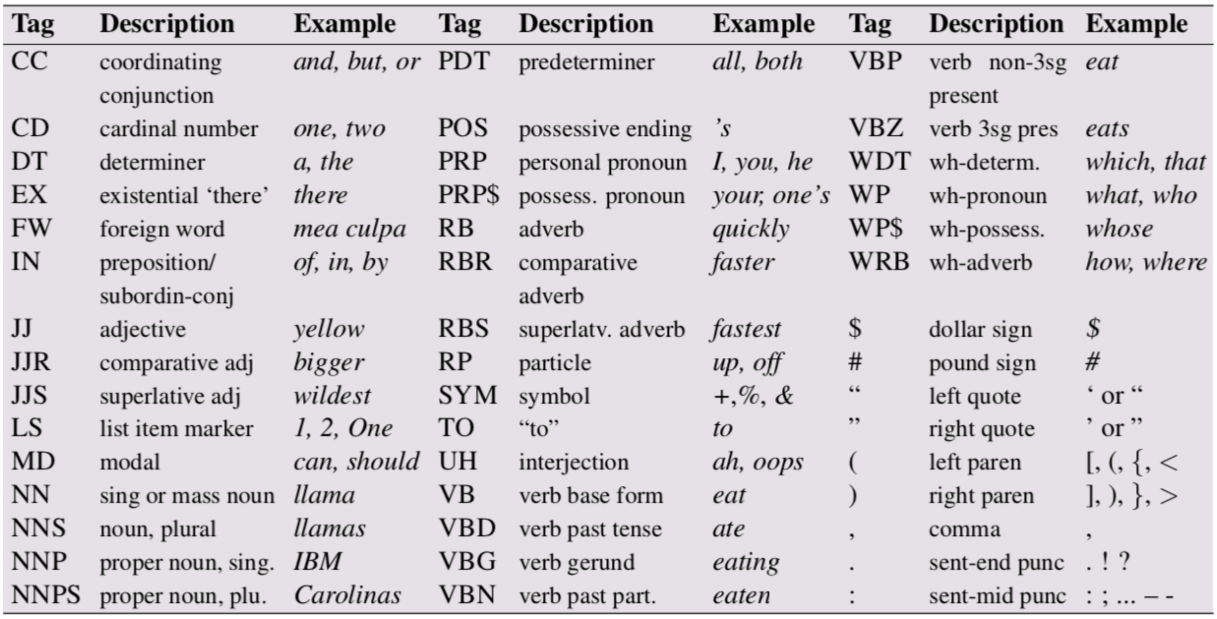
Neural language models today achieve much higher predictive accuracy than n-gram models [Jurafsky and Martin, 2019]. They can take into account much longer word windows and can generalize over similar words. Training neural language models is, however, significantly slower than training bigram or trigram models.

While neural language models have been used since the early 2000s [Bengio et al., 2003], Collobert et al. [2011] first demonstrated that embeddings can be used to represent word meanings and thus model language. Mikolov et al. [2010, 2013a,b] picked up this approach but used RNNs instead of feedforward networks. This paved the way for modern word vector models such as “word2vec” [Mikolov et al., 2015] and “GloVe” [Pennington et al., 2014]. A recent development in the field of neural language models are contextual embeddings, which combine individual word vectors into contextual vectors that represent the meaning of a whole phrase or sentence [Peters et al., 2018].

2.1.3 Part-of-Speech Tagging

Part-of-speech tagging is the process of assigning a part-of-speech (POS) label to each word in an input text. As words can have more than one possible POS, POS tagging is mainly a disambiguation task. The challenge is to choose the proper tag for the context. For unknown words, where no list of possible tags is available, features like word shape, prefixes and suffixes can be used [Brants, 2000]. Common tagsets are the 45 tags of the seminal Penn Treebank (see Section 2.1.8) shown in Figure 2.2 as well as the condensed 17 tagset proposed by the Universal Dependencies project [Nivre et al., 2016].

Earlier POS tagger implementations often use directed sequence models such as HMMs [Brants, 2000] or MEMMs [Ratnaparkhi, 1996]. The state-of-the-art Stanford POS tagger (see Section 3.1) uses a bidirectional version of a MEMM called Cyclic Dependency Network [Toutanova et al., 2003].



Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*’s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential ‘there’	*there*	PRP$	possess. pronoun	*your, one’s*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	*$*
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	*#*
JJS	superlative adj	*wildest*	SYM	symbol	+,%, &	“	left quote	‘ or “
LS	list item marker	*1, 2, One*	TO	“to”	*to*	”	right quote	’ or ”
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Figure 2.2: Penn Treebank POS tags (including punctuation)

2.1.4 Parsing

Syntactic parsing is the task of recognizing sentences and assigning a syntactic structure to them. The output of a parser depends on the chosen linguistic formalism. The two grammatical models most common in NLP are *constituency* grammars and *dependency* grammars. Both model the syntactic structure of a sentence through a directed tree that connects all words to a single root node. In the simpler and older dependency model, every word is connected to the word it directly relates to (i.e. is grammatically dependent on), with the predicate of the sentence forming the root. The more recent constituency model instead captures the individual phrases (constituents) of the sentence as well as their relations.

Just like POS tagging, syntactic parsing is challenging because of the ambiguities of natural language.

Traditionally, dependency parsing is done using transition-based techniques [Nivre, 2008; Zhang and Nivre, 2011; Goldberg and Nivre, 2012]. Chen and Manning [2014] introduced the use of neural network classifiers within such transition-based frameworks, Kiperwasser and Goldberg [2016] successfully used LSTMs for this purpose and Strubell and McCallum [2017] achieved the current state-of-the-art with CNNs.

2.1.5 Named Entity Recognition

Named Entity Recognition (NER) is the task of identifying spans of text that refer to named entities within one or more categories. These categories can be general classes like persons, locations or organizations, but also domain-specific classes like gene and protein names or financial asset classes. The task is often extended to phrases that represent numerical values, such as dates, times or percentages.

NER is usually defined as a word-by-word sequence labeling task, in which the assigned tags capture both the boundary and the entity type [Jurafsky and Martin, 2019]. To assign such tags, the raw text must first be segmented into tokens and sentences. Many NER algorithms combine lists of known entity names with rules or machine learning techniques that exploit the surrounding context of a word. According to McDonald [1996], the recognition of named entities should involve two kinds of evidence:

> **Internal** evidence is derived from within the sequence of words that comprise the name. This can be definitive criteria, such as the presence of known 'incorporation terms' ("Ltd.", "G.m.b.H.") that indicate companies; or heuristic criteria such as abbreviations or known first names often indicating people. [...] By contrast, **external** evidence is the classificatory criteria provided by the context in which a name appears. The basis for this evidence is the obvious observation that names are just ways to refer to individuals of specific types (people, churches, rock groups, etc.), and that these types have characteristic properties and participate in characteristic events. The presence of these properties or events in the immediate context of a proper name can be used to provide confirming or criterial evidence for a name's category. [McDonald, 1996]

McDonald [1996] further states that external evidence is a necessity for high NER accuracy. His argument is that in cases of referent class ambiguity, the correct entity class can only be determined by context. He also shows that predefined word lists can never cover all recognizable entities, even though they are one of the most important features in NER [Mikheev et al., 1999; Ratinov and Roth, 2009; Hoang et al., 2017; Purves et al., 2018; Jurafsky and Martin, 2019].

Table 2.1 lists a number of possible sources for internal and external evidence. The sources are categorized into *lexical and grammatical* features, which can be derived from the input text itself or preceding NER steps, and *distributional* features, which are based on corpus statistics. The table consolidates features suggested by McDonald [1996], Mikheev et al. [1999], Bender et al. [2003], Ratinov and Roth [2009], Zhang and Nivre [2011], Sutton et al. [2012], Tsai and Roth [2016], Lample et al. [2016], Purves et al. [2018] and Jurafsky and Martin [2019].

Word shape features represent a word as a pattern string that only informs about the type and case of its characters (e.g. "Xxxxx" for "Peter" or "#xxxxxx0000" for "#summer2019"). In English, word shape is single most useful feature for entity recognition

	Lexical / Grammatical	Distributional
Internal	Word shape	Word identities / embedding
	Word prefix / suffix	Prior probability
	Word lemma	Word cluster probability
	Word POS tag	
	Word patterns	
External	Word window shapes	Word window identities / embeddings
	Word window prefixes / suffixes	Sentence embedding
	Word window lemmas	Joint probabilities
	Word window POS sequence	Joint cluster probabilities
	Parse tree (dependency relations)	
	Sentence predicate	
	Sequence patterns	

Table 2.1: Possible features for NER (beyond word lists)

because proper names are almost always capital-cased [McDonald, 1996; Jurafsky and Martin, 2019]. For informal texts however, this feature is often ambivalent or misleading [Ritter et al., 2011; Mendes et al., 2011; Gelernter and Balaji, 2013].

Lemmas are the grammatically normalized forms of words, such as “go” for “went” or “corpus” for “corpora”.

Word cluster probability features do not measure the probability of an individual word within the corpus, but that of a group of similar words [Ratinov and Roth, 2009]. Words can be clustered using arbitrary similarity measures. A common strategy is Brown clustering [Turian et al., 2010], which is a predecessor of the word embedding techniques described above.

The word and sequence pattern features include hand-written (regular) expression used to identify or exclude certain words or sequences. Such patterns can identify entities like persons (“Mr. X”, two title-cased words connected by a hyphen), locations (“south of X”, preposition before proper noun) or companies (“X Inc.”) with high precision. In machine learning models such rules can be implicitly learned by the model using a combination of other provided features [Purves et al., 2018].

The size of the word window used to collect external evidence varies from system to system. Most systems use two to four words around the target word [Finkel et al., 2005; Zhang and Nivre, 2011; Honnibal and Johnson, 2015; Tsai and Roth, 2016].

Machine learning models traditionally used for NER are HMMs, Maximum Entropy models [Berger et al., 1996] and CRFs [Finkel et al., 2005]. Through the seminal work of Collobert et al. [2011] neural networks in combination with word embeddings have become increasingly popular for NER applications. Recent state-of-the-art NER systems use Bi-LSTM networks [Lample et al., 2016; Chiu and Nichols, 2016] and CNNs [Strubell et al., 2017].

2.1.6 Entity Linking

The task of Entity Linking (EL) is to associate an entity name with its representation in an *ontology* [Ji and Grishman, 2011]. In computer science, an ontology is a database that lists, defines, describes, classifies and relates entities within selected domains of discourse. EL usually happens after NER and therefore relies on its results.

An important early example of EL is the Wikify! project, that automatically enriched text with links to Wikipedia articles [Mihalcea and Csomai, 2007]. Wikipedia since became a popular ontology for EL, as it is a comprehensive, up-to-date, semi-structured and metadata-enriched collection of entities of all categories [Ling et al., 2015]. Instead of directly referring to Wikipedia articles, another common choice is to refer to DBpedia entries [Mendes et al., 2011]. DBpedia is an open, fully-structured database of entities extracted from Wikipedia and other Wikimedia projects [Lehmann et al., 2015]. The automated extraction process crawls Wikipedia's "info boxes" and models the data points into a knowledge graph. The resulting ontology spans over 500 classes, which together form a single, comprehensive taxonomy. Until 2016, an alternative to DBpedia was the collaborative Freebase project, which has been shut down in the meantime. Commercial solutions further use proprietary knowledge graphs developed within the same company (i.e. Google Cloud Natural Language, IBM Watson Natural Language Understanding, Refinitiv OpenCalais - see Section 3.3).

More recently, many publications propose joint NER and EL systems. Such systems model the recognition and linking tasks as one single mathematical problem. Sil and Yates [2013] for instance present a joint model that takes a large set of recognized names from various NER systems and a large set of candidate entity links from EL systems and then ranks all candidate-name pairs to make joint predictions. Durrett and Klein [2014] present a joint model that also performs coreference resolution together with NER and EL, using a formally structured CRF. Luo et al. [2015] use a CRF-like model that predicts both the entity type and the linked entity of a mentioned name.

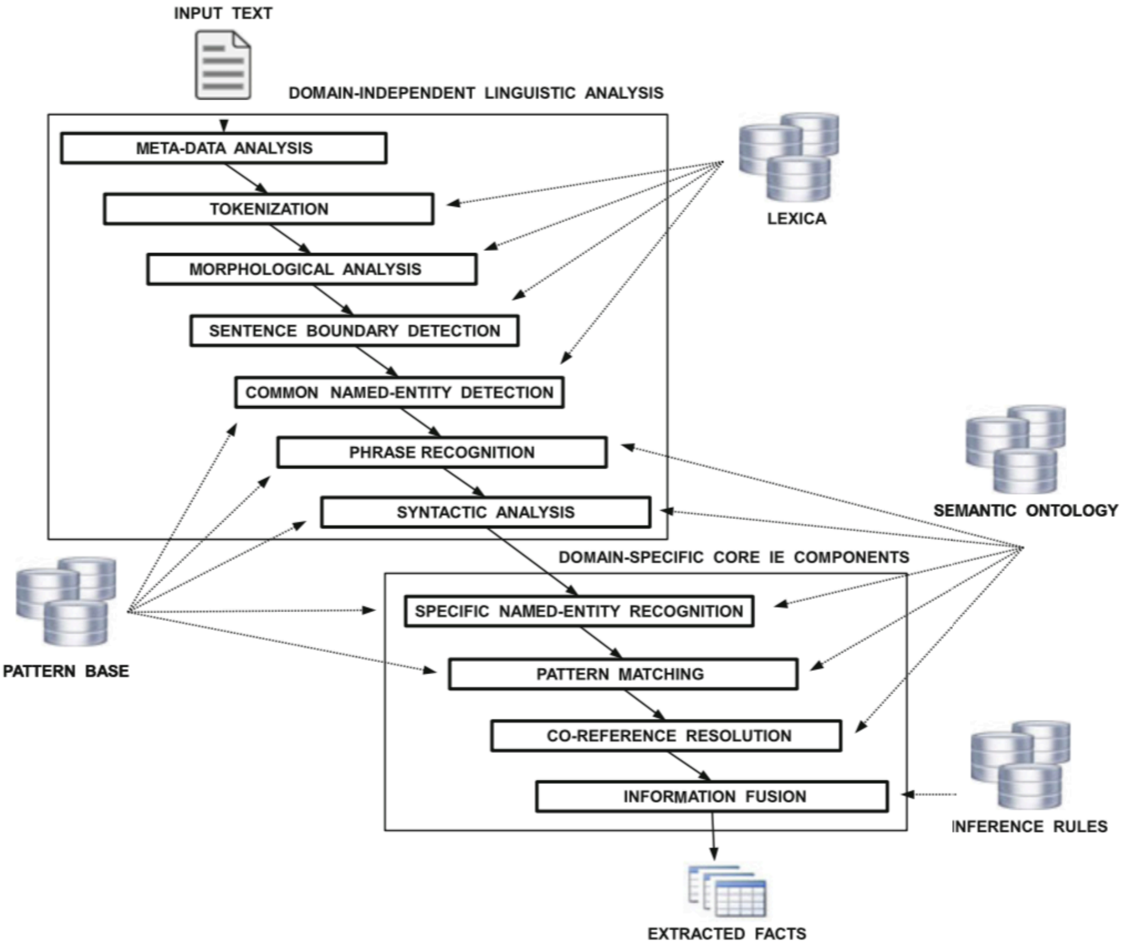


Figure 2.3: Typical architecture of an information extraction system [Piskorski and Yangarber, 2013]

2.1.7 Information Extraction

Information extraction (IE) is the task of automatically extracting structured information from unstructured text. It makes use of several NLP techniques such as parsing, NER, EL and coreference resolution. Usually IE does not pursue full-text understanding but focuses on extracting "salient facts about pre-specified types of events, entities, or relationships, in order to build more meaningful, rich representations of their semantic content" [Piskorski and Yangarber, 2013]. Such representations are used to facilitate knowledge discovery in large collections of documents.

IE usually involves identifying named entities and finding semantic relations between them. In most scenarios, additional domain-specific knowledge is required to derive structured information from the raw extraction results. Figure 2.3 shows a prototypical IE pipeline, as described by Piskorski and Yangarber [2013].

IE systems often depend on expert knowledge in the form of hand-crafted rules, patterns and templates to link and extract information [Piskorski and Yangarber, 2013; Chiticariu et al., 2013]. At the same time they rely on the performance of NLP tools, which greatly benefitted from the adoption of supervised learning techniques [Jurafsky and Martin, 2019].

For very specialized IE tasks, data sparsity can become a problem. Data sparsity describes the situation when positive training examples are so rare in the available corpora that there are too few samples for machine learning models to generalize properly [Piskorski and Yangarber, 2013].

2.1.8 Linguistic Corpora

A treebank is a parsed text corpus that annotates syntactic or semantic sentence structure. These annotations are referred to as *gold* labels, defining the output that an NLP system should ideally produce. This section briefly lists the most important corpora used in NLP and NER literature.

Started in 1992, the Penn Treebank Project provides POS tags and dependency parse trees for three major linguistic corpora [Marcus et al., 1993]. These corpora are the Brown corpus, a collection of texts from different genres published in the United States in 1961, the Wall Street Journal (WSJ) corpus, a collection of newspaper articles published in 1989, and the Switchboard corpus, a collection of transcribed telephone conversations recorded between 1990 and 1991.

The BBN Pronoun Coreference and Entity Type Corpus additionally provides named entity annotation for the Penn Treebank corpus [Weischedel and Brunstein, 2005].

The Universal Dependency Treebanks are part of the Universal Dependencies project that unified several earlier dependency tagsets [Nivre et al., 2016]. Various treebanks for over 80 languages are available. The treebanks provide lemmas, POS tags and dependency parse trees.

A corpus that captures syntax as well as various layers of semantics is the extensive OntoNotes project started in 2006 [Hovy et al., 2006]. OntoNotes uses the Penn Treebank model but adds several annotation layers on top. The latest version, OntoNotes 5.0, features the following annotation types [Pradhan et al., 2013]:

- Syntax (Penn Treebank)
- Predicate Arguments (PropBank)
- Word Sense (similar to WordNet)
- Entity Names
- Ontology References

- Coreference

Standardized NER Tasks

Most NER systems are trained, evaluated and compared on a small number of standard corpora and tasks. A corpus frequently used for this purpose is the OntoNotes corpus described above [Durrett and Klein, 2014; Honnibal and Johnson, 2015; Chiu and Nichols, 2016; Strubell et al., 2017; Khashabi et al., 2018; Apache Software Foundation, 2019]. The BNN corpus can be used accordingly. A task that many modern NER systems still use as a benchmark is the shared CoNLL 2003 Named Entity Recognition task [Tjong Kim Sang and De Meulder, 2003], compiled by the Conference on Computational Natural Language Learning (CoNLL) [Khashabi et al., 2018; Devlin et al., 2018; Apache Software Foundation, 2019]. The annotation schema used in the CoNLL tasks distinguishes between four entity types, persons, organizations, locations and other entities. Earlier NER tasks include the Entity Detection and Recognition tasks (2002-2008) published by the NIST Automatic Content Extraction (ACE) Program[1] and the MUC-6 and MUC-7 tasks published by the DARPA Message Understanding Conference (MUC) [Grishman and Sundheim, 1995; Chinchor and Robinson, 1997]. All these tasks and corpora come with detailed annotation guides, which then get adopted by the NER systems trained on those resources. With the exception of OntoNotes, all corpora mentioned here consist exclusively of newspaper articles. OntoNotes additionally includes texts from weblogs and Usenet newsgroups, as well as transcriptions of telephone calls, radio broadcasts and talk shows.

2.1.9 Summary

This section presented the central tasks, technologies and resources used in NLP to extract meaning from unstructured text. The next section will demonstrate how these techniques can be applied in the field of Geographic Information Systems to translate geographic references in unstructured text into more formal representations.

2.2 Geographic Information Extraction

The internet contains large amounts of text documents, which contain large amounts of geographic information [Hahmann and Burghardt, 2013]. Geographic Information Ex-

[1] `https://www.ldc.upenn.edu/collaborations/past-projects/ace`

traction (GIE) describes the process of extracting such information by converting it from words in natural language to formal structures suitable for automated processing. GIE is therefore a specialized form of IE.

By granting computer systems access to textual geographic information, GIE enables a wide range of applications - from intelligence [Grishman and Sundheim, 1995; Chinchor and Robinson, 1997] and social media analysis [Inkpen et al., 2015] over scientific [Weissenbacher et al., 2015] and humanitarian efforts [Middleton et al., 2013; Hoang et al., 2017; Al-Olimat et al., 2018] to graphical presentation [Amitay et al., 2004; Adams et al., 2015].

The major challenge of GIE is that most geographic references in unstructured text are under-specified and ambiguous [Purves et al., 2018]. In order to make use of the geographic information locked in unstructured text throughout, elaborate detection and disambiguation systems are needed. While the detection step involves understanding natural language, the disambiguation step requires knowledge about geographical entities. The following sections provide an overview of methods and challenges in GIE, focusing on the geographical side of the process.

2.2.1 Terminology

Following the definition of Gritta et al. [2018a], a *location* is any of the potentially infinite physical points on Earth identifiable by coordinates. A *toponym* is any named entity that labels a particular location. Within a text, toponyms referring to a specific location are also called *georeferences* [Purves et al., 2018].

Toponyms can be further separated into two groups. *Literal toponyms* are toponyms that directly refer to the place of things or events (e.g. “in England”). *Associative toponyms* instead modify otherwise non-locational terms (e.g. “British scientists”). Special types of toponyms are *demonyms*, *metonyms* and *homonyms*. Demonyms are derived from toponyms and denote the inhabitants of a country, region or city (e.g. “Londoners”). Metonyms are figures of speech where a toponym is substituted for a concept related to that location, such as a sports team or a political institution (i.e. “Manchester will face Arsenal”). Homonyms are words that are spelled like toponyms but refer to a non-locational entity such as a person or organization (e.g. “Chelsea hurt her leg”).

Toponyms are ambiguous. According to both Leidner [2007] and Batista et al. [2010] three forms of toponym ambiguity exist: (1) *referent class ambiguity* when a name is used to designate both locations and other classes of entities (“Washington” can be both a toponym or a surname); (2) *referent ambiguity* when a name refers to more than one physical location (“Washington” can be both a state and a city - see Figure 2.4); (3)

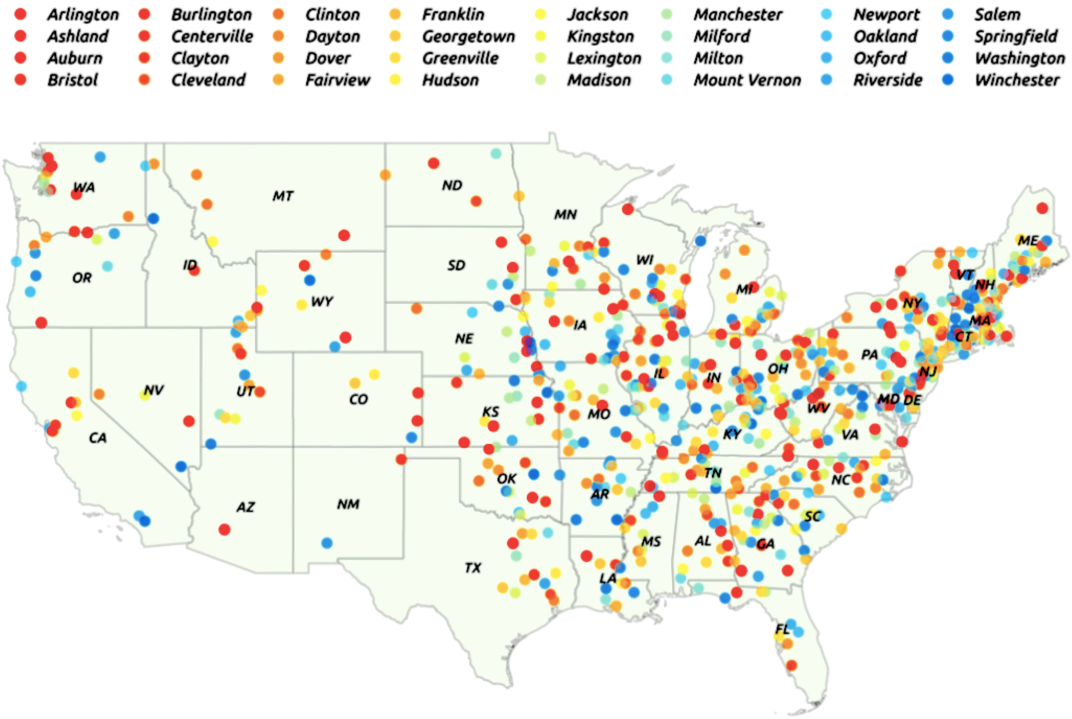


Figure 2.4: Map of the most ambiguous toponyms in the U.S. [Ju et al., 2016]

reference ambiguity when multiple names refer to the same physical location ("Washington" and "D.C." may refer to the same city). Homonyms and metonyms are examples of type (1). Amitay et al. [2004] call type (1) *geo/non-geo* ambiguity and type (2) *geo/geo* ambiguity.

For the extraction steps of coordinate information from unstructured text many different and sometimes overlapping terms exist [Purves et al., 2018]. The task of annotating all text spans that represent toponyms is called geoparsing [Batista et al., 2010; Gelernter and Balaji, 2013; Purves et al., 2018], geotagging [Sultanik and Fink, 2012; Gritta et al., 2018b], toponym detection [Inkpen et al., 2015] or toponym recognition [Leidner, 2007; Lieberman et al., 2010]. The task of assigning spatial coordinates to a detected toponym is called geocoding [Batista et al., 2010; Gelernter and Balaji, 2013; Gritta et al., 2018b], geotagging [Amitay et al., 2004], geolocation [Karimzadeh et al., 2019], grounding [Leidner, 2007; Zhang and Gelernter, 2014], toponym disambiguation [Inkpen et al., 2015] or toponym resolution [Leidner, 2007; Lieberman et al., 2010; Speriosu and Baldridge, 2013; Samet, 2014; DeLozier et al., 2015]. When referring to both tasks together, the terms georeferencing [Purves et al., 2018], geoparsing [Gritta et al., 2018b; Karimzadeh et al., 2019], geotagging [Lieberman et al., 2010] and location extraction [Hoang et al., 2017; Al-Olimat et al., 2018] have been used[1]. This paper will follow the terminology of

[1] Note that depending on the author "geotagging" could mean either the first, the second, or both steps.

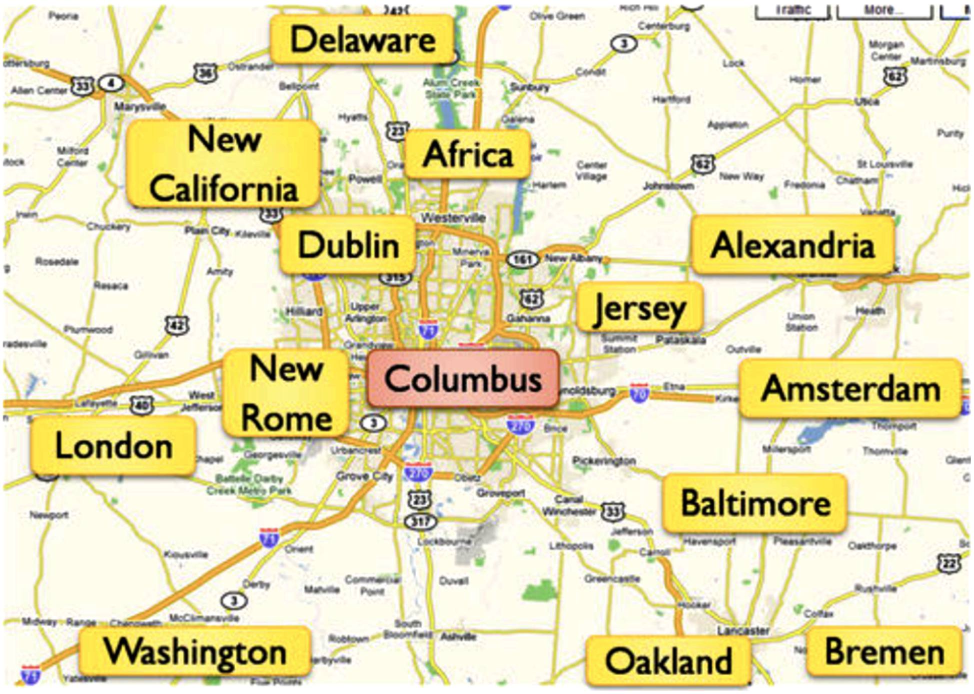


Figure 2.5: Illustration of an ambiguous local lexicon [Lieberman et al., 2010]

Leidner [2007] when talking about the individual processing steps: first *toponym recognition*, then *toponym resolution*. For the actual systems that perform these tasks, the more practical term *geoparser* (as chosen in most recent publications on the topic) will be used throughout the document.

Geographical dictionaries that contain toponyms together with associated metadata such as coordinates, place type or population size are called *gazetteers*. The term is also used for entity name lists in general [Jurafsky and Martin, 2019]. To refer to the gazetteers inherently used by humans, Lieberman et al. [2010] use the term *lexicons*. They describe a *global* lexicon, which is more or less shared between all speakers of a language, and *local* lexicons which are specific to a group of speakers in a distinct geographic region. Figure 2.5 illustrates how toponyms in a local lexicon (here for the region of Columbus, Ohio, U.S.) can overlap with the common global lexicon. Lieberman et al. [2010] assume that "in most cases, the local lexicon supersedes the global lexicon".

Although similar in nature, geoparsing should be separated from *geocoding*. Geocoding is the task of resolving a distinct textual representation of a location to a coordinate [Goldberg et al., 2007]. The main difference to geoparsing is that for the geocoding task, it is already known that the full input text represents one single location. This means that recognition is not necessary, but also that no textual context for disambiguation is available. If no additional information is provided, geocoders are therefore expected to retrieve the most relevant location candidate for the given input. Many geocoders further allow to specify a reference location for geocoding requests (e.g. the widely used

Google Geocoding API). In geoparsing terms, geocoding could therefore be described as the subtask of retrieving the default sense for a recognized toponym (see Section 2.2.3).

2.2.2 Toponym Recognition

The first step of the geoparsing is toponym recognition. Toponym recognition can be understood as a subdiscipline of NER, where only location entities are of interest. In turn, most techniques feasible for general-purpose NER can also be used for toponym recognition (see Section 2.1.5). Among systems designed specifically for toponyms detection, CRFs, HMMs, Decision Trees and Maximum Entropy models are commonly used [Purves et al., 2018]. Especially CRFs, which are being used in state-of-the-art NER systems, have been successfully adopted for location-specific NER tasks [Batista et al., 2010; Gelernter and Zhang, 2013; Inkpen et al., 2015].

As toponym recognition is a subdiscipline of NER, recognition systems make use of the same internal and external evidence presented in Section 2.1.5 (see Figure 2.1). Gazetteers are often found to be the single most useful feature to detect toponyms [Mikheev et al., 1999; Inkpen et al., 2015; Malmasi and Dras, 2015; Hoang et al., 2017]. Unlike other named entities, whose class can often be inferred from sentence context, location references tend to offer few reliable clues for contextual classification [Mikheev et al., 1999]. Their recognition therefore requires additional sources of knowledge, like gazetteers. Mikheev et al. [1999] also show that the most useful gazetteers are those that "contain very common names, names which the authors can expect their audience already to know about".

Among geoparsing systems operating on a global scope, the most commonly used gazetteer is GeoNames[1]. GeoNames is a public geographical database that contains over 25 million geographical names with metadata such as coordinate, elevation and population (used by [Lieberman et al., 2010; Tobin et al., 2010; Gelernter and Balaji, 2013; Speriosu and Baldridge, 2013; Middleton et al., 2013; Weissenbacher et al., 2015; Inkpen et al., 2015; Malmasi and Dras, 2015; Berico Technologies, 2016] etc.). In addition to this central source of knowledge, geoparsing literature mentions a wide range of national, regional, governmental, domain-specific and self-compiled gazetteers. Map databases such as OpenStreetMap (OSM), Google Maps, Foursquare and Natural Earth can further serve as gazetteers at street level [Middleton et al., 2013; DeLozier et al., 2015; Al-Olimat et al., 2018; Ji et al., 2016]. A standardized framework of country names, county codes, subdivision names, subdivision codes and the relationships among them is provided through ISO 3166[2]. For supervised systems that are trained on linguistic corpora, the corpora

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

[2] https://www.iso.org/iso-3166-country-codes.html

themselves serve as implicit gazetteers. For such systems, it is important that the learning approach generalizes well to allow the recognition of toponyms not mentioned in the training data [Lieberman et al., 2010].

The simplest method to use gazetteers for toponym recognition is to search for exact matches of known toponyms. Such gazetteer lookup approaches were common in early geoparsing systems [Amitay et al., 2004], but have several limitations. First and foremost, gazetteers are always a compromise between precision and recall. Due to referent class ambiguity (geo/non-geo), a larger gazetteer will also always produce more false positives [Amitay et al., 2004]. To covers all towns, for example, a gazetteer has to include "To" in Myanmar, "Of" in Turkey, "Us" in France, "Metro" in Indonesia and "Book" in Louisiana (U.S.). To avoid false positives, gazetteers can be limited to unambiguous (or less ambiguous) toponym. But this will necessarily reduce comprehensiveness and cause false negatives [Purves et al., 2018]. Another problem with gazetteers is the malleability of toponyms [Leidner, 2007; Purves et al., 2018]. Especially in less formal texts (e.g. social media), toponyms are often abbreviated, contracted or colloquialized [Malmasi and Dras, 2015; Hoang et al., 2017; Al-Olimat et al., 2018]. Gazetteers meanwhile tend to elongate names [Al-Olimat et al., 2018]. The official name of Germany in GeoNames is, for example, "Federal Republic of Germany". Many systems therefore use fuzzy or partial matching to enrich their gazetteers with auto-generated variants of toponyms [Gelernter and Balaji, 2013; Malmasi and Dras, 2015; Berico Technologies, 2016; Ji et al., 2016; Al-Olimat et al., 2018; Karimzadeh et al., 2019]. However, this has been shown to further exacerbate the ambiguity problem [Al-Olimat et al., 2018].

A method that can achieve very high precision is the use of hand-crafted linguistic rules. Such rules include word patterns such as "city of X" or "south of X" but also word lists with generic components of toponyms (e.g. "Street", "School") [Rauch et al., 2003; Middleton et al., 2013; Gelernter and Balaji, 2013; Malmasi and Dras, 2015; Al-Olimat et al., 2018]. But since it is very hard to design a comprehensive framework of rules that covers all types of location references, most authors use recognition rules only in combination with other high-recall methods [Batista et al., 2010; Pasley et al., 2007; Qin et al., 2010].

Although word lists, gazetteers and hand-crafted rules traditionally play a large role in NER, some recent general-purpose NER systems achieve state-of-the-art performance without using any of those (e.g. Stanford NER and spaCy). Their assumption is that with enough training data, lexical features like word shape, embeddings and POS tags can provide sufficient evidence to reliably recognize entities. Several recent publications show that this approach also works well for toponyms [Purves et al., 2018; Gritta et al., 2018b; Karimzadeh et al., 2019]. In hybrid recognition systems, general-purpose NER systems can further compensate some of the problems inherent to gazetteer lookup approaches, and vice versa [Mikheev et al., 1999; Gelernter and Balaji, 2013; Daiber et al., 2013].

An approach unique to toponym recognition is to used geographical language models to decide whether a phrase represents a location. Such models capture the prior and joint probabilities of words and word groups to refer to a geographical point or region [Al-Olimat et al., 2018; Gritta et al., 2018b]. Normally, supervised machine learning techniques are used to train such models.

2.2.3 Toponym Resolution

After having decided which names refer to locations, the goal of toponym resolution is to find the actual locations the recognized toponyms refer to. Toponym resolution is therefore a subdiscipline of the EL task described in Section 2.1.6. The final output of toponym resolution systems are usually either geographic coordinates or references to geographic ontologies whose entries are associated with coordinates. The coordinates represent a specific location in the physical world (thus the term “grounding”). Since many toponyms can refer to more than one physical location, the main focus of toponym resolution systems is to resolve referent ambiguities (geo/geo). The resolution of reference ambiguities - cases when two toponyms refer to the same location - is part of the task, but becomes trivial once all toponyms are resolved correctly.

A wide range of publications exist on the topic. While early work mostly relied on gazetteers and hand-built rules, recent literature focuses more on supervised and unsupervised machine learning models (see Section 3.2 in the next chapter). Models that have been used for toponym resolution include SVMs [Qin et al., 2010; Zhang and Gelernter, 2014; Melo and Martins, 2015], Decision Trees [Wang et al., 2005; Lieberman and Samet, 2012], Naive Bayes [Daiber et al., 2013], Beam Search [Ji et al., 2016] and more recently CNNs [Gritta et al., 2018b].

The initial question that guides most work in toponym resolution is the following: “How do humans resolve toponyms, and how can computers replicate this?”. Utilizing their knowledge and experience of the real-world as well as evidence from within the document, humans usually disambiguate encountered toponyms without problems [Qin et al., 2010; Ju et al., 2016]. According to Leidner [2007], there are two reasons for this. On one hand, author and reader usually share some extra-linguistic context in which common toponyms have preferred senses. This intuition has been formalized in the lexicon theory of Lieberman et al. [2010]. On the other hand, authors implicitly adhere to Grice’s Cooperative Principle, which requires the writer to provide as much disambiguation information as their audience can possibly require [Grice, 1975; Leidner, 2007]. Both arguments lead researchers to the assumption that, in general, the information to correctly disambiguate a toponym is available - either in the document itself or in the form of common knowledge.

This assumption lead to several disambiguation heuristics that attempt to leverage the available information. The next sections will discuss disambiguation heuristics frequently used in geoparsing literature.

One-sense-per-referent Heuristic

In linguistics, a coherent structured group of sentences is called a discourse [Hobbs, 1985]. Gale et al. [1992] formulated the theory that during one discourse, the same name will always refer to the same entity. Leidner [2007] applied this theorem to toponym resolution, calling it the *one-sense-per-referent heuristic*. It suggests that identical location names within one document should always be resolved to the same physical location. This heuristic has since become a standard practice for toponym resolution. All systems presented in this thesis make use of this heuristic, most without explicitly mentioning it.

Default Sense Heurisitc

The *default sense heuristic* suggests to always prefer the toponym sense that is most prominent to the largest number of speakers, such as the French capital sense for the toponym “Paris” [Purves et al., 2018]. In practice, the default sense is usually approximated by choosing the gazetteer entry with the largest population or the highest hierarchy level (i.e. countries over cities) [Leidner, 2007]. Default sense heuristics often serve as a baseline for toponym resolution evaluation (e.g. [Speriosu and Baldridge, 2013; Gritta et al., 2018a; Karimzadeh et al., 2019]) or as a fallback strategy when no other heuristics can be applied (e.g. [Leidner, 2007; Rauch et al., 2003; Batista et al., 2010; Weissenbacher et al., 2015]). Gritta et al. [2018a] demonstrate that for some corpora always choosing the default sense can correctly resolve up to 95% of all toponyms. This rate, of course, depends on the level of ambiguity and locality of the toponyms within a given corpus.

Indicator Words Heuristics

The assumption behind *indicator word heuristics* is that the words before and after a detected toponym can provide important evidence for resolution. This evidence may be embedded in other toponyms, non-locational entity names or regular words like nouns, verbs and adjectives. While the use of other toponyms for disambiguation is an established practice in geoparsing literature [Rauch et al., 2003; Leidner, 2007], the use of non-toponym words has only become feasible with machine learning techniques and large training corpora (see below). The next two sections will describe the two traditional ways in which other toponyms are used for disambiguation. The section thereafter will describe techniques that use statistical language models to predict the correct toponym sense. All three approaches are variants of indicator word heuristics.

Spatial Minimality Heurisitc

The intuition behind the *spatial minimality heuristic* is that co-occurring toponyms should be resolved to locations that minimize the geographic distance between them. While Leidner [2007] generally assumes that toponyms appearing together within a document refer to geographically related locations, Lieberman et al. [2010] notes that this is only true if the document has a single geographic focus, which is not always the case. Karimzadeh et al. [2019] demonstrate that for documents with global geographic scope, the spatial minimality assumption can lead to suboptimal results. Nevertheless, spatial minimality heuristics are very common in toponym resolution literature (e.g. [Leidner, 2007; Lieberman et al., 2010; Kamalloo and Rafiei, 2018; Karimzadeh et al., 2019]).

Ontological Similarity Heuristic

The *ontological similarity heuristic* (or *ontological distance heuristic* [Purves et al., 2018]) can be described as a text-based version of the spatial minimality heuristic. Instead of finding resolution sets that minimize geographical distance, the ontological similarity heuristic seeks to minimize the distances between resolution candidates in an ontological entity graph [Lieberman and Samet, 2012]. In practice, that entity graph is the global administrative hierarchy of continents, countries, administrative districts and cities, as defined in ISO 3166 and modelled by GeoNames [Amitay et al., 2004; Lieberman and Samet, 2012]. Batista et al. [2010] present a mathematical formalism based on this approach which they call “semantic similarity”. Lieberman and Samet [2012] propose a variant of this heuristic that uses adaptively sized context windows to dynamically adjust their system to the toponym density of input documents.

Language Model Heurisitcs

Language model heuristics make use of geo-statistical language models to select the most likely toponym sense. Such models capture the probabilities of words or toponyms to refer to certain location entities or coordinate regions. These probabilities are derived from large geo-annotated corpora, usually with machine learning techniques. A popular source of geo-annotated documents is Wikipedia [Speriosu and Baldridge, 2013; Melo and Martins, 2015; DeLozier et al., 2015; Gritta et al., 2018b]. By combining the probabilities of the recognized toponyms with those of nearby indicator words, the most probable entity or coordinate region can be inferred.

The less complex variant of the heuristic is to only calculate probabilities for toponyms or toponym groups [Melo and Martins, 2015; Santos et al., 2015; Gritta et al., 2018b]. The computationally more expensive variant is to model the geospatial distribution profiles of all words in the language [Speriosu and Baldridge, 2013; DeLozier et al., 2015; Ju et al., 2016; Gritta et al., 2018b]. The advantage of the latter is that non-toponyms can contribute to the disambiguation decision, i.e. words like "capital", "White House" or "Wizards"[1] can serve as strong indicators that the city sense of "Washington" is meant, not the much larger U.S. state [Speriosu and Baldridge, 2013]. The disadvantage of this variant is that it requires large amounts of training data to compensate for data sparsity.

The language models described here are all machine learning classifiers. They use word vectors[2] as input features to predict which class a toponym belongs to. Language model approaches therefore transform the toponym resolution task into a classification problem. This is done by dividing the surface of Earth into equally sized grid cells that can serve as output classes. An important factor in this transformation is the side length of those cells. Smaller cells mean more classes, which complicate model training. Larger cells reduce resolution accuracy. The achievable resolution further depends on the density and granularity of the location references in the training data. In practice, used cell sizes are between 0.5 and 2 degrees latitude and longitude [Melo and Martins, 2015; Santos et al., 2015; Gritta et al., 2018b]. This translates to a resolution accuracy of 50-200 km, which is in stark contrast to the more or less exact center coordinates obtained from GeoNames. For this reason, DeLozier et al. [2015] implement an optional gazetteer mapping step in their resolution system, which retrieves the most fitting gazetteer entry for the given name and grid cell.

2.2.4 Corpora

A big challenge within toponym recognition (but also in toponym resolution and IE in general) is to find or generate enough annotated training data to achieve good generalization [Piskorski and Yangarber, 2013; Lieberman et al., 2010; Zhang and Gelernter, 2014; Purves et al., 2018; Gritta et al., 2018a]. This section presents a range of public geo-annotated corpora suitable for training and evaluation of toponym recognition and resolution systems. The list only covers English corpora.

The *Local-Global Lexicon* (LGL) is a collection of 588 news articles collected from 78 local, geographically distributed newspapers [Lieberman et al., 2010]. All toponyms were annotated manually. Some articles were specifically chosen to include non-default-sense

[1] The Washington Wizards are a well-known basketball team in the U.S.

[2] Gritta et al. [2018a] additionally use "MapVec" vectors that represent the geographical meaning of a word.

toponyms (e.g. local newspapers from towns called Paris in the United States). According to Gritta et al. [2018b], most of the toponyms refer to entities below the size of U.S. states. Approx. 16% of all annotated toponyms are not assigned to coordinates at all. The LGL corpus is one of the few corpora in geoparsing literature that has been used by more than one group of researchers to train and evaluate their systems [DeLozier et al., 2015; Santos et al., 2015; Kamalloo and Rafiei, 2018; Gritta et al., 2018b; Karimzadeh et al., 2019].

The *Wikipedia Toponym Retrieval* (WikToR) corpus is a collection of 5000 Wikipedia articles programmatically annotated with GeoNames entries [Gritta et al., 2018c]. The corpus intentionally contains many toponyms that are highly ambiguous, i.e. have multiple entries within GeoNames, such as Santa Maria (26 entries), Santa Cruz (25), Victoria (23), Lima (19) and Santa Barbara (19). The authors note that the corpus should not be used to measure recall, because for every article only occurrences of a single toponym is annotated, which is the toponym that the article is about.

GeoVirus is a dataset of news articles covering global disease outbreaks and epidemics [Gritta et al., 2018b]. It was constructed from free WikiNews and annotated manually. 2,167 toponyms are assigned with coordinates and Wikipedia page URLs. Buildings, points of interest, street names and rivers are not annotated.

TR-News is a corpus of 118 local and global newspaper articles from 2018, annotated with data from GeoNames [Kamalloo and Rafiei, 2018]. Similar to WikToR, the corpus contains many ambiguous toponyms to put more focus on disambiguation performance.

GeoWebNews is a manually annotated dataset of 200 articles from globally distributed news sites [Gritta et al., 2018a]. These contain 2,720 toponyms which are annotated with either GeoNames identifiers or coordinates. According to the authors, 8% of the annotated toponyms (like facilities, buildings, street names, park names, festivals, universities and other venues) require either an extra source of knowledge (other than GeoNames), a self-compiled list of businesses and organization names or even human-like inference to resolve correctly.

Based on the number of ambiguous toponyms, Gritta et al. [2018b,a] assign the following geoparsing difficulties:

- GeoVirus: easy to moderate
- GeoWebNews: easy to moderate
- LGL: moderate to hard
- WikToR: very hard

As described in Section 2.2.3, several authors have further used Wikipedia articles as a source of geographically annotated text. This is possible because many Wikipedia articles are tagged with geographical coordinates. This subset of articles is sometimes called "GeoWiki" in lack of an official name [Speriosu and Baldridge, 2013]. Other researches are exploring unsupervised learning methods to allow training on unlabelled data [Ji et al., 2016].

2.2.5 Summary

This section defined the task of geoparsing and described its two main subtasks, toponym recognition and toponym resolution. It further gave an overview of important techniques, models, features and heuristics used to accomplish these two tasks. This knowledge serves as a theoretical framework for the remainder of this thesis.

Chapter 3

Previous and Related Work

This chapter provides an overview of previous work in the fields of Named Entity Recognition and Geographic Information Extraction relevant for this thesis. The first part presents general-purpose NER systems frequently used for toponym recognition. The second part briefly summarizes the development of geoparsing systems and approaches over the last two decades. The last part introduces a selection of popular commercial Entity Linking solutions, which will later be used as baselines for evaluation.

3.1 Named Entity Recognition

Many geoparsers make use of general-purpose NER tools for toponym recognition (compare Section 3.2). This section presents five freely available NLP frameworks that have previously been chosen for this task, starting with long-standing rule-base solutions and ending with modern systems based on deep neural networks.

Initiated by the University of Sheffield in 1995, the General Architecture for Text Engineering (GATE) is a suite of open-source NLP tools [Cunningham et al., 2014]. The modular architecture allows the creation of custom IE pipelines with reusable components and a powerful rule engine. The software package comes with a pre-built IE pipeline named ANNIE (A Nearly-New IE System). This pipeline makes use of GATEs standard sentence segmentation, tokenization, POS tagging and coreference resolution components. For NER it additionally includes large sets of gazetteers and hand-crafted rules [Cunningham et al., 2002]. Recognized entity types are locations, persons and organizations, as well as addresses and numerical types (date, money, percentage, etc.).

Apache OpenNLP is an open-source NLP toolkit based on machine learning[1]. It supports

[1] `https://opennlp.apache.org`

several common NLP tasks, such as tokenization, sentence segmentation, POS tagging, NER, chunking, parsing, and coreference resolution. The included NER module is called Apache OpenNLP Name Finder. Internally the module uses the maximum entropy approach presented by Bender et al. [2003]. The list of recognized entity types is similar to that of GATE, including locations, persons and organizations. OpenNLP offers separate pre-trained models for each entity type[1]. Although not stated explicitly, mentions in the manual imply that the OntoNotes 4 corpus as well as the MUC-6, CoNLL 2002 and CoNLL 2003 NER tasks were used to train those models [2].

Stanford CoreNLP is a collection of linguistic analysis tools developed by the Stanford Natural Language Processing Group [Manning et al., 2014]. It includes tools for POS tagging, NER, parsing, coreference resolution, IE, sentiment analysis and bootstrapped pattern learning. The NER module, named Stanford NER, uses CRFs to recognize location, person and organization names [Finkel et al., 2005]. The CRF models were trained on corpora of newspaper articles (CoNLL, MUC-6, MUC-7 and ACE) without the use of explicit gazetteers.

CogComp NLP is a collection of NLP libraries developed by the Cognitive Computation Group at the University of Pennsylvania [Khashabi et al., 2018]. The software contains tools for tokenization, POS tagging, chunking, NER, lemmatization, dependency and constituency parsing and other tasks. For NER CogComp uses the Illinois Named Entity Tagger (NET), which is based on averaged perceptron classifiers [Ratinov and Roth, 2009; Tsai and Roth, 2016]. The software comes with a 4-entity-type model trained on the CoNLL task and an 18-entity-type model trained on the OntoNotes corpus (types include geopolitical entities, locations, organizations and faculties). The system uses gazetteers extracted from Wikipedia (collectively containing over 1.5 million names) and language models derived from unlabeled text [Ratinov and Roth, 2009].

spaCy is an NLP framework based on CNNs and neural language models [Honnibal and Johnson, 2015; Explosion AI, 2019]. It supports tokenization, POS tagging, dependency parsing, lemmatization and NER. Both the dependency parser and the named entity recognizer use the incremental transition-based approach proposed by Nivre [2008] and refined by Goldberg and Nivre [2012]. The algorithm iterates over each token and labels them using lexical features derived from a small context window, among them contextual word embeddings of all words in the window. The recognized entity types are persons, nationalities or religious or political groups (“NORP”), faculties, locations, geopolitical entities, products, events, works of arts and laws. For English NER, three pre-trained

[1] `http://opennlp.sourceforge.net/models-1.5/`

[2] `https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html` (accessed December 2019)

models are provided, which differ in size due to dimensions of the embedding model. All spaCy models are trained on the OntoNotes 5 corpus. While spaCy's current NER algorithm has not been published in a paper yet, evaluation results published by the author (Honnibal) claim[1] that on the OntoNotes 5 corpus the system performs significantly better than the one used in CogComp ([Ratinov and Roth, 2009]) and the CRF-based system by Durrett and Klein [2014]. It supposedly achieves accuracies comparable to the state-of-the-art CNN systems presented by Strubell et al. [2017] and Chiu and Nichols [2016].

3.2 Geoparsing

This section provides an overview of past and current approaches to geoparsing. It is structured as a timeline of ideas and techniques, ranging from the earliest systems presented over 15 years ago to the state-of-the-art approaches used today. This knowledge is indispensable to analyze and understand the design decisions in modern geoparsing systems and comprises many building blocks for the design of new solutions.

One of the earliest geoparsing systems is MetaCarta, presented by Rauch et al. [2003]. It employs a purely confidence-based disambiguation approach for both recognition and resolution. Toponyms are detected using **gazetteer lookup** combined with linguistic pattern and anti-pattern matching. For disambiguation default sense and spatial minimality **heuristics** are used. Amitay et al. [2004] present a similar geoparsing system called Web-a-Where, which also uses gazetteers lookup for recognition and a default sense heuristic for resolution. Wang et al. [2005] additionally uses an ontological similarity heuristic that uses the administrative hierarchy and population data to generate decision trees for disambiguation.

Observing the bad performance of MetaCarta and Web-a-Where for non-default toponym senses, Lieberman et al. [2010] present a new geoparsing system that uses global and **local lexicons** for toponym resolution. The system's main toponym recognition method is gazetteer lookup, using GeoNames as reference gazetteer. Additionally, Stanford NER and hand-written rules are used. The toponym resolution algorithm combines multiple features, among them the distance from the previously know document location and the presence in the global and local lexicons. The global lexicon is a regular gazetteer containing geopolitical entities with populations over 100,000. The local lexicon is a pre-compiled region-specific gazetteer that captures the toponym vocabulary and default senses of a selected area. Part of the presented system is an algorithm to extract such local lexicons from annotated collections of documents that are representative for the

[1] `https://spacy.io/universe/project/video-spacys-ner-model` (accessed December 2019)

region of interest. The authors show that the use of local lexicons significantly improves resolution performance. In a follow-up publication, Lieberman and Samet [2012] further improve the toponym resolution performance of their system by adopting basic machine learning techniques. They train a classifier based on decision trees to learn the optimal combination of different features. They further make the size of the context window for disambiguation features adjustable, allowing users to tweak the system for different textual domains.

A similar system from the same time that is still used as evaluation baseline today is the Edinburgh Geoparser [Tobin et al., 2010; Alex, 2017]. By making the gazetteer interchangeable and integrating an online platform to host **custom gazetteers**, it also allows the user the narrow the linguistic scope of the system. The authors for example use gazetteers specific to Great Britain or to historical toponyms. GeoNames serves as the default gazetteer. The geoparser uses a custom rule-based recognition system called LT-TTT2 to find toponyms [Grover and Tobin, 2006]. For each recognized name resolution candidates are retrieved from the selected gazetteer. These candidates are ranked using the heuristics suggested by Leidner [2007], namely default sense, spatial minimality and ontological similarity. For each heuristic, every candidate is assigned a value between zero and one. The weighted sum of these values determines the final score of the candidate.

With the rise of social media platforms, many publications attempted to adopt existing geoparsing approaches to the less formal texts posted on such mediums, mostly focusing on **Twitter**. Middleton et al. [2013] pick up the idea of global and local lexicons originated by Lieberman et al. [2010] and further extend it by adding street-level data obtained from the Google Places API and OpenStreetMap into their local gazetteers. Global names are obtained from GeoNames and a similar database provided by the U.S. government. For toponym recognition, the input text is first tokenized using an NLP toolkit. The system then tries to match spans of words from the text with entries in the previously compiled gazetteer, preferring larger entities over smaller ones. As social media texts often contain abbreviated and contracted toponyms, the prepared gazetteers are augmented with various alternative forms of the obtained names, including hashtag versions. Malmasi and Dras [2015] use a similar recognition approach, but limit the n-gram matching to noun phrases extracted via NLP techniques. They use the Stanford CoreNLP tools for tokenization, POS tagging and constituency parsing. From the constituency parse trees they obtain all noun phrases, which are then matched against a GeoNames gazetteer as well as several sets of hand-crafted rules and word lists used to detect addresses, points of interest and spatial expressions (e.g. “25 km North of Beijing”). Zhang and Gelernter [2014] use SVMs to resolve toponyms in tweets, combining geographic features obtained from GeoNames with Twitter metadata such as location and biographical text of the author.

SVMs are also used to resolve toponyms in other systems. Qin et al. [2010] present a system that iterates between recognition and resolution, using SVMs to find and select optimal resolution candidates. The candidates are ranked using default sense, ontological similarity and indicator word heuristics. The latter uses search result statistics from Google to estimate co-occurrence probabilities. Melo and Martins [2015] use a hierarchy of SVMs to assign coordinates to documents. The layers of the hierarchy correspond to increasingly smaller cells on a global grid. Each SVM is trained on Wikipedia articles that cover locations in the region they represent. By traversing the SVM hierarchy from top to bottom, new documents can be assigned to the grid cell that best fits the mentioned toponyms.

For custom toponym recognition models, **CRF**s are a common choice. Batista et al. [2010] recognize toponyms using a CRF model adapted from Cohen [2004] and trained on smaller Portuguese corpora. The model uses a combination of lexical, gazetteer and pattern-based features. For resolution candidates retrieved from the same gazetteer are disambiguated using an ontological similarity heuristic. Gelernter and Zhang [2013] present a toponym recognition system for tweets that combines gazetteer matching with rule-based parsers for street and building names and a self-trained CRF (additionally Stanford NER is used for English documents). The CRF model uses both lexical and gazetteer features, derived from a context window of +/-3 words around the toponym. A manually annotated corpus of approx. 4500 tweets is used as training data. The gazetteer matching approach utilizes POS patterns to find lookup candidates and then matches them against a GeoNames gazetteer. Inkpen et al. [2015] train their own CRF to recognize toponyms in tweets, using both gazetteer and lexical features, including POS tags obtained from a Twitter-specific NLP tool. The model does not only recognize toponyms, but also classifies them into city, provinces/state and country names. The authors report that both the gazetteer and the lexical features contribute significantly to the recognition performance. Gritta et al. [2018a] also train a toponym-specific CRF to evaluate toponym recognition on their corpus.

Weissenbacher et al. [2015] extend the GATE/ANNIE pipeline with own gazetteers and **linguistic rules** in order to recognize and resolve toponyms. They first replace the included gazetteer with a more comprehensive one derived from GeoNames gazetteer. They further provide a set of rules and black-lists to filter out acronyms, names of people, names of organizations, other common names and demonyms. For disambiguation, they use default sense and spatial minimality heuristics, as well as domain-specific metadata available for their use case. A similar approach is pursued by the OpenSextant project[1].

[1] `https://opensextant.github.io`

With the availability of large training corpora and more powerful hardware, the performance of **general-purpose NER** tools saw significant improvements in the last years, making them increasingly attractive as stand-alone toponym recognition solutions. The Cartographic Location And Vicinity Indexer (CLAVIN), developed by Berico Technologies [2016], uses the OpenNLP Name Finder to recognize toponyms. For resolution the system retrieves candidates from GeoNames and the applies default sense and (optionally) spatial minimality heuristics to disambiguate them. The geoparsing system proposed by Kamalloo and Rafiei [2018] uses Stanford NER for toponym recognition, GeoNames as gazetteer and a spatial minimality heuristic that incorporates the estimated geographical document scope. The GeoTxt system by Karimzadeh et al. [2019] uses six different NER algorithms for toponym recognition in tweets, among them the Illinois Named Entity Tagger, Stanford NER, OpenNLP Name Finder and GATE/ANNIE. The resolution strategy is similar to the one employed by CLAVIN, using a search engine to index, retrieve and rank GeoNames candidates. For disambiguation default sense, spatial minimality and ontological similarity heuristics can be used. It is further possible to extend the search engine queries with custom ranking criteria.

Large corpora also enabled the development of **statistical language models** for toponym resolution. One of the first systems using such a model is the Wikipedia Indirectly Supervised Toponym Resolver (WISTR) presented by Speriosu and Baldridge [2013]. Unlike the other systems considered in their paper, WISTR uses a toponym co-occurrence model based on Logistic Regression classifiers and trained on geo-annotated Wikipedia articles. Each classifier can disambiguate a single toponym present in the training data, using features derived from a context window of +/-20 words around the toponym. The system uses the OpenNLP Name Finder for toponym recognition and GeoNames to provide resolution candidates. DBpedia Spotlight, a tool that automatically annotates mentions of DBpedia resources in unstructured text (not only toponyms), works similarly [Mendes et al., 2011; Daiber et al., 2013]. For entity name recognition it uses OpenNLP Name Finder combined with two sets of syntactic patterns. The first pattern detects all sequences of capitalized tokens. The second pattern detects noun phrases, prepositional phrases and multi-word units. After filtering detected names for false positives using a set of weighted features, resolution candidates are disambiguated using an indicator word heuristic implemented using Naive Bayes classifiers. The classifiers are trained on Wikipedia articles, learning the absolute probabilities of every entity as well as the conditional probabilities of every entity given a name and surrounding words. The TopoCluster system described by DeLozier et al. [2015] (again specialized on toponyms) goes a step further and resolves toponyms by combining the spatial distribution profiles of all words in a context window around a recognized toponym. The pre-calculated profiles model the prior probabilities of every word in the language to be biased to any cell in a global 0.5 by 0.5 degrees grid of the Earth. The distribution profiles are derived from geo-annotated Wikipedia articles. Ju et al. [2016] use two separate language models to resolve toponyms in tweets. One focuses on related entity names obtained from DBpedia. The other models the bias of arbitrary words towards different toponym senses and is trained on geo-annotated Wikipedia articles using an unsupervised learning algorithm. Al-Olimat et al. [2018] present a tool (LNEx) that detects toponym boundaries in tweets using region-specific n-gram language models that predict whether a group of words represent a toponym, based on occurrence counts in a gazetteer. Like Middleton et al. [2013], Al-Olimat et al. [2018] use a gazetteer that contains alternative, abbreviated and contracted forms of names obtained from GeoNames.

Ji et al. [2016] present a **joint model** for toponym recognition and resolution of toponyms in tweets. The model is restricted to a preselected, geographically contained set of location candidates retrieved from Foursquare. A Beam Search algorithm is used to find an optimal double-layer of annotations for the input text. The first annotation layer denotes which groups of tokens together represent toponyms. The second layer associates token groups with location candidates. This annotation scheme makes it possible to perform recognition and resolution simultaneously in a joint search space. Location-related features, lexical features and gazetteers are used for prediction. The lexical features are derived from a context window of +/-5 words and include word shape and POS tags. The gazetteer feature checks if a token or token group is present in a pre-generated gazetteer derived from the Foursquare candidate set. Location-related features include the similarity of detected toponym and location candidate name, the popularity of the candidate on Foursquare and other features based on candidate metadata. To compare complete sets of annotations, the system uses a spatial minimality heuristic and a heuristic that penalizes annotation overlaps. The optimal feature weights are learned from a manually annotated corpus of 900 tweets that mention locations in the gazetteer. To “alleviate the dearth of labeled data” an unsupervised learning algorithm with access to up to 15,000 unlabelled tweets containing gazetteer matches is implemented as well [Ji et al., 2016].

In recent years, **deep neural networks** have been used for toponym resolution. Santos et al. [2015] present a CNN that combines an exhaustive set of manually generated lexical and geographic features, effectively applying ontological similarity and spatial minimality heuristics. CamCoder, a CNN-based toponym resolution system presented by Gritta et al. [2018b], uses a novel geographic word vector model (MapVec) to embed toponyms. MapVec produces sparse, spatial vectors that explicitly model the geographic distributions of a toponym, just as word embeddings represent the linguistic meaning of words. Additionally, a linguistic language model is used to embed non-toponym words. Both models were trained on geo-annotated Wikipedia articles. Disambiguation is done using input features derived from a context window of +/-200 words. By utilizing both a linguistic and a geographical language model, the system achieves a performance improvement over previous approaches [Gritta et al., 2018b].

3.3 Commercial Entity Linking Solutions

This section presents commercial Entity Linking services that can be used for geoparsing. It will be shown that some of these services match or even outperform the state-of-the-art systems from academia presented above. When comparing street-level geoparsing performance, those services should not be overlooked.

Refinitiv Intelligent Tagging, formerly Reuters OpenCalais[1], is an NLP-based entity tagging service operated by Thomson Reuters and referenced in several geoparsing publications [Lieberman and Samet, 2012; Kamalloo and Rafiei, 2018]. The service supports document and word-level tagging and performs both entity recognition and resolution. As reference ontology the service uses the companies own knowledge graph called "PermID Linked Data Graph", which uses unique permanent identifiers called PermIDs to link entities and metadata. The long list of entity and relation types reflects the company's focus on strategic and financial analysis (e.g. companies, financial instruments, acquired-by, competitor-of, etc.) but also includes geographical entities types such as continents, countries, regions, cities, provinces/states, natural features, organizations and facilities. According to the online documentation[2], the facility class includes man-made physical entities such as universities, hospitals, courts, embassies, consulates, radio and TV stations as well as theme and amusement parks, but no settlements, farms, parking lots, parks or streets. Coordinates are provided for resolved countries, provinces, states and cities.

The Google Cloud Natural Language API is a service offered by Google as part of their Google Cloud AI & Machine Learning suite. The service performs various natural language understanding tasks including sentiment analysis, entity analysis, entity sentiment analysis, content classification and syntax analysis, using Google's own pre-trained NLP models. The models are not public and no details are provided. Recent publications and blog posts[3] by Google researchers suggest that they are based on the Transformer architecture, which uses deep, bidirectional, unsupervised language models [Vaswani et al., 2017; Devlin et al., 2018; So et al., 2019].

[1] As of October 2018 the former Financial and Risk business division of Thomson Reuters is known as Refinitiv. OpenCalais now refers to a service level within the Intelligent Tagging API.

[2] Available for registered users at `https://developers.refinitiv.com/open-permid/calais-tagging-restful-api/docs` (accessed December 2019)

[3] `https://ai.googleblog.com/search/label/Natural%20Language%20Processing` (accessed December 2019)

The entity analysis endpoint of the Natural Language API performs NER and EL. Each recognized entity is linked to an item in Google's own ontology called the Google Knowledge Graph, as well as a Wikipedia article (in some cases). The Google Knowledge Graph supports 20 entity classes, among them places, organizations and persons. Google does not provide geographic coordinates for Knowledge Graph items, therefore toponym resolution can only be done indirectly by retrieving the coordinate of the linked Wikipedia article, if available.

IBM Watson Natural Language Understanding is the NLP service of the Watson AI platform from IBM. The web service can extract concepts, entities, keywords, and sentiment. It also links recognized entities to DBpedia entries. The list of entity types includes locations, companies and facilities[1]. No implementation details are disclosed.

[1] https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-entity-type-systems (accessed December 2019)

Chapter 4

Gap Analysis

Before designing a new system for street-level geoparsing, it is important to ensure that existing systems do not already provide sufficient solutions for the given problem. To allow an objective comparison, the first section defines the functional requirements for the task. The next section then presents the sample text that is used to test the geoparsing capabilities of existing systems. With this sample text, additional texts and results from previous work it is then shown that neither of the selected systems from academia and industry can meet the specified requirements. The last section of the chapter discusses why the tested systems are incapable of street-level geoparsing and points out possible approaches to overcome their limitations.

4.1 Target Functionality

Chapter 1 introduced and motivated the task of street-level geoparsing. To determine whether existing systems are already capable of this feat, the stated task description must be translated into a testable set of requirements. The following list specifies which references a street-level geoparsing system should be able to extract:

- Preconditions
  - The input text is in English
  - The input text is grammatically correct
- Recognition Requirements
  - Names of geopolitical entities should get recognized
  - Names of streets should get recognized
  - Names of points of interest should get recognized, including

        - Landmarks
        - Public institutions
        - Public spaces
        - Tourist attractions
    - Names that refer to multiple locations can be ignored (i.e. chains)
    - Names that do not refer to physical locations can be ignored
    - Demonyms can be ignored
    - Non-canonical toponym forms can be ignored
    - Historical toponyms can be ignored
    - House numbers in addresses can be ignored

- Resolution Requirements
    - Recognized toponyms should be resolved to either
        - geographic coordinates or
        - entries in a reference ontology that provide shape data
    - For geopolitical entities and streets, ontology entries are preferred

The extraction of full addresses with house numbers is not required for two reasons. On one hand, addresses are not strictly named entities or toponyms (but include those). Because of this NER-based geoparsing methods fail to recognize and resolve them as such. On the other hand, full addresses are unambiguous by design, and therefore can be recognized using regular expressions (e.g. by GATE/ANNIE) and resolved using standard geocoding APIs. Consequently, address parsing should be considered a separate task and solved using separate methods. The toponyms included in addresses - especially street names - should get extracted nonetheless.

Non-canonical toponyms have also been excluded because they require specialized methods and knowledge to resolve. The extraction of non-canonical references is certainly "nice-to-have", but is not considered a mandatory feature for street-level geoparsing at this point.

The chosen set of requirements is of course tailored to the specific use case of this thesis, which is to link literary content to landmarks in cities and encourage exploration. The requirements therefore focus on physical locations that people might want to visit. References to groups of people, non-locational entities, historical or redundant places are less relevant in this scenario. For other applications, the target functionality might differ.

The sample text presented in Chapter 1 (reproduced in Figure 4.1) was constructed as a test suite for the requirements specified here. The text contains instances of all three groups, in varying degrees of difficulty for both recognition and resolution. In total, the text contains ten toponyms, ranging from city to street level. All refer to locations in the city of New York.

> *The rapper Jay-Z loves his hometown New York. He grew up in Brooklyn, but now lives in Tribeca and can be seen driving expensive cars down Broadway and 8th Street. His old apartment was located on the corner of State Street and Flatbush Avenue in Bed-Stuy. In his song ”Empire State of Mind” he mentions the Statue of Liberty, the Empire State Building and the World Trade Center.*

Figure 4.1: Sample text with toponyms in New York City

The first toponym, “New York”, sets the geographic scope for all subsequent toponyms. This first toponym already represents an ambiguity problem, as New York can refer both to the State New York, one of the United States, and its capital, New York City (NYC). New York State has a bigger population than NYC, but still the NYC sense is usually considered to be the default sense of the word.

To provide heuristics with sufficient context to choose the correct toponym sense, the next two toponyms are names strongly linked with administrative sub-divisions of NYC: “Brooklyn” and “Tribeca”. Brooklyn is the most populous borough of NYC and Tribeca is a neighborhood in Lower Manhattan, which is the cities central borough. The three entities are therefore on different levels of the administrative hierarchy. Later in the text, another neighborhood in NYC is mentioned: “Bed-Stuy”, which is an alternative name for Bedford–Stuyvesant, a neighborhood in Brooklyn. This is an example of a non-canonical toponym form, and therefore not expected to be resolved. But since the name is covered by all reference ontologies of the tested systems, it can reveal which systems actually consider alternative names.

These toponyms all refer to geopolitical entities and are covered by common geoparsing gazetteers such as GeoNames. For the remaining toponyms, this is not necessarily true. “Broadway”, “8th Street”, “State Street” and “Flatbush Avenue” all refer to streets in Manhattan and Brooklyn. The first three names are highly ambiguous, as streets with those names exist in many English-speaking cities. Broadway can alternatively refer to the Broadway theater, a famous group of theater and musical venues located along that street. The semantics of the text imply the street sense, but the theater sense will be accepted here as well. “8th Street” poses a syntactic challenge because it does not start with an uppercase letter. Toponym recognition approaches that rely too heavily on the capitalization feature might not be able to detect it. “State Street” is hard to resolve because it exhibits both geo/geo and geo/non-geo ambiguity. The street does not only share its name with many larger streets in other cities, but also with a large international bank.

The last three toponyms refer to points of interest in NYC. All three are (or were) famous landmarks, but prove difficult for geoparsing. “Statue of Liberty” is challenging in three ways. First syntactically, as the two title-cased proper nouns are separated by a lower-case preposition. This makes name boundary detection non-trivial. Then grammatically, because all three components are common English words and “of” functions as a stop word in many NER systems. And finally geographically, because “Liberty” is the name of a small town in New York State. “World Trade Center” again is ambiguous, as the World Trade Center Association operates multiple World Trade Centers all around the globe[1]. Technically it is also a historical toponym, because the World Trade Center referenced here sadly does not exist anymore. But all reference ontologies of the tested systems contain entries for that name at the correct location. The only difficulty with “Empire State Building” is that it is a composite word. Interestingly, one of the systems tested below did recognize only the “Empire” part and resolved it to the Wikipedia page of the Roman Empire[2].

The examples in the text demonstrate why street-level geoparsing is challenging. The next section compares toponym recognition and resolution results of state-of-the-art geoparsing systems using this and other texts. Of course, a single example can never prove a systems ability to fulfill a whole set of requirements. But it can be used to analyze the capabilities and limitations and select approaches that work best for the selected examples.

4.2 Status Quo

The goal of this section is to determine whether currently available solutions for toponym recognition and resolution are capable of street-level geoparsing. For both disciplines, the same methodology is applied. First, through broad literature analysis, the techniques and systems currently most feasible for the task are identified. Proprietary solutions that are used as baselines by other researchers or are found to perform well are considered as well. These systems are then tested using input texts designed to reveal their limitations in a street-level geoparsing scenario.

[1] `https://en.wikipedia.org/wiki/List_of_World_Trade_Centers`

[2] Compare annotation results of DBpedia Spotlight `https://www.dbpedia-spotlight.org/demo/` (reproducible December 2019)

NER Tool	P	R	F1
CogComp	**0.91**	**0.69**	**0.79**
Stanford	0.90	0.62	0.74
GATE/ANNIE	0.86	0.62	0.72
Stanford (case insensitive)	0.90	0.59	0.72
OpenNLP	0.67	0.29	0.40

Table 4.1: Recognition results for tweets reported by Karimzadeh et al. [2019]

4.2.1 Toponym Recognition

As shown in Chapter 3.2, many recent geoparsing systems delegate the toponym recognition task to general-purpose NER tools. Common choices for such tools are Stanford NER [Lieberman et al., 2010; Gelernter and Zhang, 2013; Ling et al., 2015; Kamalloo and Rafiei, 2018; Karimzadeh et al., 2019], OpenNLP Name Finder [Speriosu and Baldridge, 2013; Daiber et al., 2013; Berico Technologies, 2016; Karimzadeh et al., 2019], the Illinois Named Entity Tagger [Karimzadeh et al., 2019], spaCy NER [Gritta et al., 2018b] and GATE/ANNIE [Weissenbacher et al., 2015; Karimzadeh et al., 2019]. With the exception of GATE/ANNIE, all of these use machine learning models trained on large linguistic standard corpora.

Karimzadeh et al. [2019] evaluate the performance of six general-purpose NER systems for toponym recognition in social media using a corpus of approx. 6000 manually annotated tweets [Wallgrün et al., 2018]. Their results are reproduced in Table 4.1 (P for precision, R for recall, F1 for F1-Score). Even under difficult conditions (no case information, abbreviations, slang) some of the general-purpose NER tools achieve F1-Scores over 70%. The good performance of CogComp is being attributed to the large Wikipedia gazetteers used to train the models of the Illinois Named Entity Tagger. A notable finding is that GATE/ANNIE works significantly better than OpenNLP under these conditions and is almost on par with Stanford NER.

Gritta et al. [2018a] also compare the toponym recognition performance of NER tools using their GeoWebNews corpus. Table 4.2 reproduces their results. Google Cloud Natural Language and spaCy NER were chosen because they performed best in initial experiments. To study how models specialized on toponym recognition compete against general-purpose solutions, they further train a custom CRF model using the corpus itself as training data. The Edinburgh Geoparser is included for reference. The authors find that Google Cloud Natural Language achieves the highest precision, but the self-trained model achieves the best F1-Score. Unfortunately, the trained model has not been published.

NER Tool	P	R	F1
Custom CRF model	0.90	**0.87**	**0.89**
Google Cloud Natural Language	**0.91**	0.77	0.83
spaCy	0.82	0.69	0.75
Edinburgh Geoparser	0.81	0.52	0.64

Table 4.2: Recognition results for the GeoWebNews corpus reported by Gritta et al. [2018a]

Based on these previous studies, the following tools and configurations have been chosen for closer evaluation:

- **spSM** - spaCy NER using the small "en_core_web_sm" model
- **spLG** - spaCy NER using the large "en_core_web_lg" model
- **St7C** - Stanford NER using the "english.muc.7class" model
- **St4C** - Stanford NER using the "english.conll.4class" model
- **St3C** - Stanford NER using the "english.all.3class" model
- **GCNL** - Google Cloud Natural Language
- **Il4C** - Illinois Named Entity Tagger using the CoNLL model
- **Il18C** - Illinois Named Entity Tagger using the OntoNotes model
- **Open** - OpenNLP Name Finder using only the "en-ner-location" model

GATE/ANNIE is not included because its potential for street-level geoparsing can be easily estimated by reviewing their gazetteers. Below city level, those gazetteers are extremely sparse and show no inherent structure [1]. Considering that the few location-specific rules only apply to gazetteer matches and that only matched names will be annotated as locations, the system is obviously incapable of street-level geoparsing.

Table 4.3 shows the results of the selected systems for the sample text described above. It can be seen that geopolitical entities are recognized in most cases, but points of interest and street names only occasionally. Sights are often annotated as organizations or faculties, which is valid in some cases. St4C does not recognize "8th Street" while detecting other street names, possibly due to reliance on title case. The annotations produced by OpenNLP Name Finder suggest that the model does not generalize well

[1] `https://github.com/GateNLP/gateplugin-ANNIE/tree/master/src/main/resources/resources/gazetteer` (accessed December 2019)

Toponym	spSM	spLG	St7C	St4C	St3C	GCNL	Il4C	Il18C	Open
New York	+	+	+	+	+	+	+	+	+
Brooklyn	+	+	+	+	+	+	+	+	+
Tribeca	+	+	+	+	+	+	+	-	+
Bed-Stuy		+	+	+	+	+	+	+	+
Empire State Bldg.	+	-	-	-	+	+	+	+	
World Trade Center	-			-		-	+	-	
Statue of Liberty	-			-		+	+		*
Broadway				-	+	+	-	-	+
8th Street						+	-	-	
State Street				+		+	-	-	
Flatbush Ave			+	+	+	+	-	-	+
+ location; - organization or faculty; * partial annotation									

Table 4.3: Recognition results for the sample text

from the toponyms presented in the training corpus. Google Cloud Natural Language is the only system that correctly recognizes all toponyms (as mentioned above, the World Trade Center is also an organization). A street-level geoparsing system that opts for a general-purpose NER system thus either has to rely on proprietary solutions or implement additional methods to detect and classify toponyms.

I was born in Tweelsford, but when I turned 15 my family moved to Penserworth, a little town 40 miles north of Opalsburgh. The next airport was 200 miles away in Cloustering, and the only time I ever left Landria was for my 21st birthday, when we visited distant relatives in Karningshire, Mudlington.

Figure 4.2: Sample text with fictional toponyms

A method that can be used to explore in how far NER systems use lexical clues is to replace real toponyms with invented toponyms. Figure 4.2 shows a sample text which contains fictional location names that do not appear in any common gazetteer or training corpus. For a human reader, the text still provides sufficient clues to understand that all names must refer to geopolitical entities. While some could be detected using pattern matching, others require a semantic understanding for correct classification. Table 4.4 shows the results of the selected NER systems for this text. The results suggest that neural systems trained without explicit gazetteers (spaCy, Stanford) are indeed capable of recognizing toponyms based on external evidence, at least in some cases. As gazetteer-centric approaches quickly become unfeasible for smaller-scale entities, this capability

Toponym	spSM	spLG	St7C	St4C	St3C	GCNL	Il4C	Il18C	Open
Tweelsford	+	+	+	+	+	+		-	+
Penserworth	+	*	+		+		+	-	
Opalsburgh		+		+	+			-	
Cloustering	+	+	+	+	+				
Landria	+		+	+	+			+	
Karningshire	+	+	+	+	+		+	+	
Mudlington		-	+	+	+		+	+	
+ location; - organization; * labelled as NORP									

Table 4.4: Recognition results for a text with fictional toponyms

becomes more relevant. The bad performance of Google Cloud Natural Language, which performed best on the text with real toponyms, can be explained by its reliance on the Google Knowledge Graph, which obviously does not cover invented names (all toponyms got recognized, but only one was classified as location).

4.2.2 Toponym Resolution

After exploring the capabilities and limitations of toponym recognition systems, this section will focus on toponym resolution.

A comprehensive comparison of toponym resolution performances can be found in Gritta et al. [2018b]. They measure the toponym resolution performance of six state-of-the-art geoparsing systems over three corpora - LGL, WikToR (WIK) and GeoVirus (GEO). Table 4.5 reproduces their results. “Population” refers to a baseline system that resolves each toponym to the GeoNames candidate with the largest population. The CamCoder and TopoCluster systems included in this comparison are not feasible for street-level geoparsing because of their low accuracy. The Edinburgh Geoparser, CLAVIN and GeoTxt all achieve good results for some of the corpora. The used Accuracy@161km metric measures the percentage of resolutions whose error distance to the gold coordinate was below 161 kilometers. A detailed discussion of this metric follows in Section 6.2.

Unfortunately, the GeoTxt online service was taken offline during work on this thesis and the published source code comes without any documentation[1]. Its resolution strategy is however quite similar to that of CLAVIN and Kamalloo and Rafiei [2018], with most of the differences catering to the idiosyncrasies of social media text [Karimzadeh et al.,

[1] `https://github.com/geovista/GeoTxt` (accessed December 2019)

Geoparser	Accuracy@161km		
	LGL	WIK	GEO
CamCoder	**0.76**	**0.65**	**0.82**
Edinburgh Geoparser	**0.76**	0.42	0.78
Population	0.70	0.22	0.80
CLAVIN	0.71	0.16	0.79
GeoTxt	0.68	0.18	0.79
TopoCluster	0.63	0.26	-

Table 4.5: Resolution results reported by Gritta et al. [2018b]

2019]. The performance of these systems should therefore serve as good approximations for the capabilities of GeoTxt.

Kamalloo and Rafiei [2018] demonstrate that their TopoResolver system can outperform commercial solutions like Reuters OpenCalais (now Refinitiv OpenCalais) and the Google Cloud Natural Language API on certain corpora. It uses GeoNames as reference ontology and combined heuristics for disambiguation, which mirrors the current standard architecture in geoparsing literature. As the source code is freely available[1], the system can serve as an indicator for the street-level geoparsing performance of this approach.

The Google Cloud Natural Language API, DBpedia Spotlight and Refinitiv OpenCalais were included because they recently served as baseline for other authors [Al-Olimat et al., 2018; Kamalloo and Rafiei, 2018; Gritta et al., 2018a]. The IBM Watson Natural Language Understanding API and the geoparser.io web service, both not yet mentioned in geoparsing literature (to my best knowledge), were included because they performed well in initial tests. Geoparser.io is a commercial geoparsing service launched in 2016. Implementation details or source code are not available, but according to their website the service uses GeoNames as gazetteer[2]. Google Cloud Natural Language, IBM Watson Natural Language Understanding, Refinitiv OpenCalais and DBpedia Spotlight all recognize named entities (not only toponyms) and link them to ontology items. As these ontologies provide coordinates for location-type entities, all four systems can be used for geoparsing (see Section 3.3).

In summary, the following systems were chosen:

- **DBPS** - DBPedia Spotlight [Daiber et al., 2013]

[1] `https://github.com/ehsk/CHF-TopoResolver`

[2] `https://geoparser.io/docs.html` (accessed December 2019)

- **gpio** - geoparser.io
- **GCNL** - Google Cloud Natural Language API
- **IBMW** - IBM Watson Natural Language Understanding
- **Edin** - Edinburgh Geoparser (GeoNames) [Alex, 2017]
- **ROpC** - Refinitiv OpenCalais
- **CLAV** - CLAVIN [Berico Technologies, 2016]
- **Kama** - TopoResolver [Kamalloo and Rafiei, 2018]

Table 4.6 shows how these systems perform on the sample text presented earlier (Figure 4.1). Google Cloud Natural Language again delivers the best results. The service correctly recognizes all toponyms except "World Trade Center" as locations and manages to resolve all but the street names (although all of them have dedicated Wikipedia articles). Most other services do not recognize street-level entities at all or resolve them to unrelated entities of larger scale. DBpedia Spotlight further exhibits problems with detecting name boundaries. As expected, Geoparser.io, CLAVIN and TopoResolver only resolve to GeoNames entries, and thus resolve street-level toponyms either not at all, or falsely to locations listed in their gazetteer. Their correct resolutions of "Flatbush Avenue" is only due to the random coincidence that GeoNames contains an entry for the "Flatbush Avenue Terminal" which is more or less located on that avenue. CLAVIN further resolved the toponym "Liberty", which was already recognized incorrectly by OpenNLP Name Finder, to a town somewhere in the state of New York. DBpedia Spotlight resolved the same incorrect match to the philosophical concept of liberty, illustrating the problem of referent class ambiguity. DBpedia Spotlight did also link "Empire" to the Roman Empire and "State Street" to an entry about Route 89, a highway in the U.S. The partial matches show the limitations of the grammatical rules used for entity name recognition. Overall, it becomes clear that none of the considered systems fulfills the requirements formulated in Section 4.1.

In the sample text about New York, most references can be resolved using the default sense heuristic, provided that the word boundaries have been detected correctly. Since New York is one of the largest and most famous cities on Earth, toponyms associated with prominent locations in this city tend to "default" to those locations. The text in Figure 4.3 was constructed to determine how well the selected systems perform for non-default toponym senses. All toponyms in the text refer to more than one physical location. "Paris" and "Odessa" are especially challenging because they have strong default senses located far away from the locations referred to in the text (in France and Ukraine, respectively). From the toponyms "Texas", "Dallas" and "US", that are all used in the default sense, it is still possible to infer the correct locations for all toponyms. The first ambiguous

Toponym	DBPS	gpio	GCNL	IBMW	Edin	ROpC	CLAV	Kama
with exact GeoNames entry								
New York	+	+	+	+	+	+	+	+
Brooklyn	+	+	+	+			+	+
Tribeca	+	+	+	-		-	+	+
Bed-Stuy	+	+	+	-			+	+
Empire State Bldg.	*		+	+	+	-		+
Broadway	+		+		-		-	-
Statue of Liberty	*		+			-	*	
without exact GeoNames entry								
World Trade Center	+		+	+		-		
8th Street			-					
State Street	-	-	-					
Flatbush Avenue	+	+	-				+	
+ correct location; - wrong location; * partial annotation								

Table 4.6: Resolution results for the sample text

toponym of the sample text ("Paris") is intentionally placed *before* the toponyms that provide context for disambiguation, as this is a pattern often found in natural language (e.g. address fragments). Also, "US" was intentionally used in the abbreviated form.

Paris is the county seat of Lamar County, Texas. It is only a 2h drive away from Dallas, the ninth most populous city in the US. Six hours west of Dallas lies Odessa, which has 2014 been ranked the third-fastest growing small city in the US. With a population of ∼100,000 Odessa is about four times as large as Paris.

Figure 4.3: Sample text with fictional location references

Table 4.7 shows the toponym resolution results for the intentionally ambiguous text. As all tested systems use the one-sense-per-referent heuristic, each toponym is only listed once in the table. All systems did recognize Paris as a toponym, but only two managed to resolve it to the correct location in Texas. Most other systems selected the default sense. IBM Watson instead chose Paris, Illinois (U.S.) and also resolved "Dallas" to Dallas, West Virginia (U.S.). The TopoResolver system linked "US" to the city of Os in Norway and "Lamar County" to Lamar, Mississippi (U.S.), but still managed to correctly resolve both "Odessa" and "Paris", which was accomplished by no other system. Overall, each system exhibited unique strengths and weaknesses, but none fully solved the task.

Toponym	DBPS	gpio	GCNL	IBMW	Edin	ROpC	CLAV	Kama
Default Senses								
US		+	+	+	+	+		-
Texas	+	+	+	+	+	+	+	+
Lamar County		+	-	+	+	-	-	-
Dallas	+	+	+	-		+	+	+
Non-Default Senses								
Paris	-	-	-	-	-	+	-	+
Odessa	-	+	+	-	+	-		+
+ correct location; - wrong location								

Table 4.7: Resolution results for the ambiguous text

In summary, none of the tested geoparsing systems could correctly resolve all toponyms in either example. The next section will analyze what causes these systems to perform badly for certain cases of toponyms and ambiguity, and what strategies could be used to improve their performance.

4.3 Limitations

Generalizing from the individual errors observed above, this section outlines the systematic limitations of current state-of-the-art geoparsing system in regard to street-level toponyms and offers explanations and solutions. After having determined both what is needed and what is currently possible in the previous sections, this final segment analyzes the gap between the two and points out approaches to narrow that gap.

4.3.1 Entity Class Ambiguity

One drawback of using general-purpose NER systems for toponym recognition is that these systems were not optimized to recognize all toponyms but to achieve a good balance of precision and recall over all named entities. And since the annotation guidelines of standard NER tasks and corpora (see Section 2.1.8) usually do not allow alternative annotations for a single entity name, NER systems are forced to pick the type that fits best and disregard evidence for other types, even if support for them is significant. Lieberman et al. [2010] argue that this necessarily leads to many misclassifications. And the results of Gritta et al. [2018a] and Karimzadeh et al. [2019] presented above indeed show that while achieving high precision, the evaluated systems also consistently suffer low recall. The results for the sample texts confirm this, with many toponyms classified as organizations, faculties or even persons. As name boundary detection is usually not an issue in grammatically correct texts (compare results in Chapter 6), the low recall measured by Gritta et al. [2018a] and Karimzadeh et al. [2019] can only stem from entity class errors. The performance differences between models with different sets of entity types observable in Table 4.3 further suggest that a larger number of entity types leads to more misclassifications.

From this characteristic of general-purpose NER tools emerges a difficult design decision for toponym recognition systems: Which entity types should be considered as possible toponyms? One option is to include entity types that toponyms frequently get confused for, such as organizations, faculties or persons. The other option is to only consider names explicitly labeled as locations. Including additional entity types necessarily leads to additional false positives, which could bias resolution heuristics. But excluding them necessarily leads to false negatives, which could potentially represent important disambiguation clues.

One possible solution for this dilemma is to use a "reclassification" gazetteers to catch entities which have been labeled as non-locations by NER but actually do refer to locations. This approach partially dismisses the classification decisions of the NER tool and assumes that ontological knowledge about geographic entities allows a more accurate classification than machine-learned language models. This seems plausible because the classification problem is already much simpler at this stage. There are further cases of referent class ambiguity that require external knowledge to resolve correctly [Mikheev et al., 1999]. A reclassification gazetteer can contribute this knowledge.

Another possible strategy is to treat the recognized location entities as high-confidence findings and then resort to other methods to recognize potentially missed toponyms. Since most modern NER systems were trained on newspaper articles from large publishers, they usually perform well for globally significant geopolitical entities [Lieberman et al., 2010]. Methods to recognize and resolve the remaining, less prominent entities will be discussed in the following sections.

4.3.2 Ontology Coverage

Toponym resolution systems can only resolve toponyms that are present in their reference ontology. The geoparsers assessed here all use GeoNames as gazetteer. Most of the EL systems use Wikipedia (or DBpedia, which is derived from Wikipedia). These two ontologies are also the most common among the publications presented in Chapter 3 (GeoNames: [Lieberman et al., 2010; Gelernter and Zhang, 2013; Speriosu and Baldridge, 2013; Zhang and Gelernter, 2014; Weissenbacher et al., 2015; Inkpen et al., 2015; Malmasi and Dras, 2015; Ju et al., 2016; Al-Olimat et al., 2018; Karimzadeh et al., 2019], Wikipedia: [Melo and Martins, 2015; DeLozier et al., 2015; Gritta et al., 2018b]).

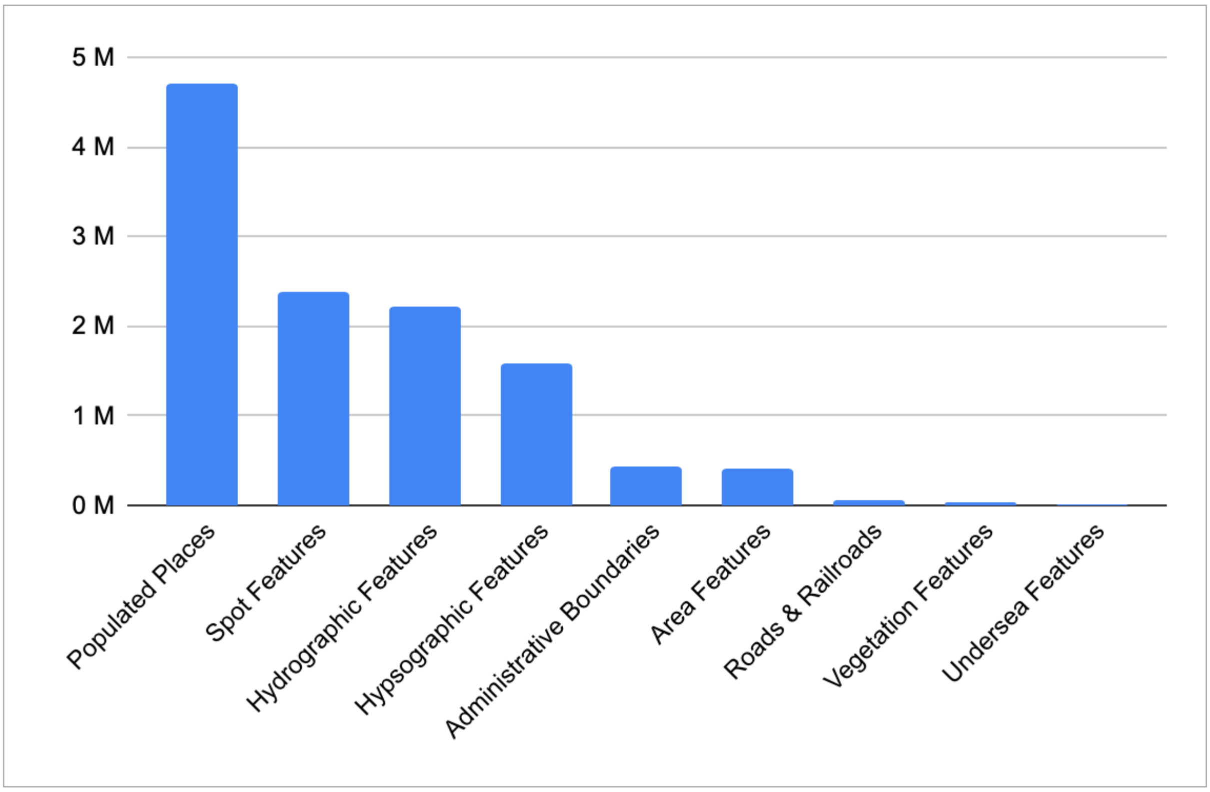
The GeoNames gazetteer contains almost twelve million entries, among them administrative boundaries, populated places, designated areas, facilities, roads and natural features in all sizes and forms. Its extensive taxonomy includes feature types as granular as bus stops, gates and steps. The number of actual entries for these types is however vanishingly small, especially for countries other than the United States. As visible in Figure 4.4, the majority of entries are populated places (cities, towns, villages, etc.), spot features (farms, churches, hotels, schools, etc.), hydrographic and hypsographic features (geological features of water and land). More than half of the ∼2.4 million listed spot features are located in the United States. Only 0.1% of all listed names (∼15,000) are actual roads. Hence, GeoNames cannot be considered a street-level ontology.

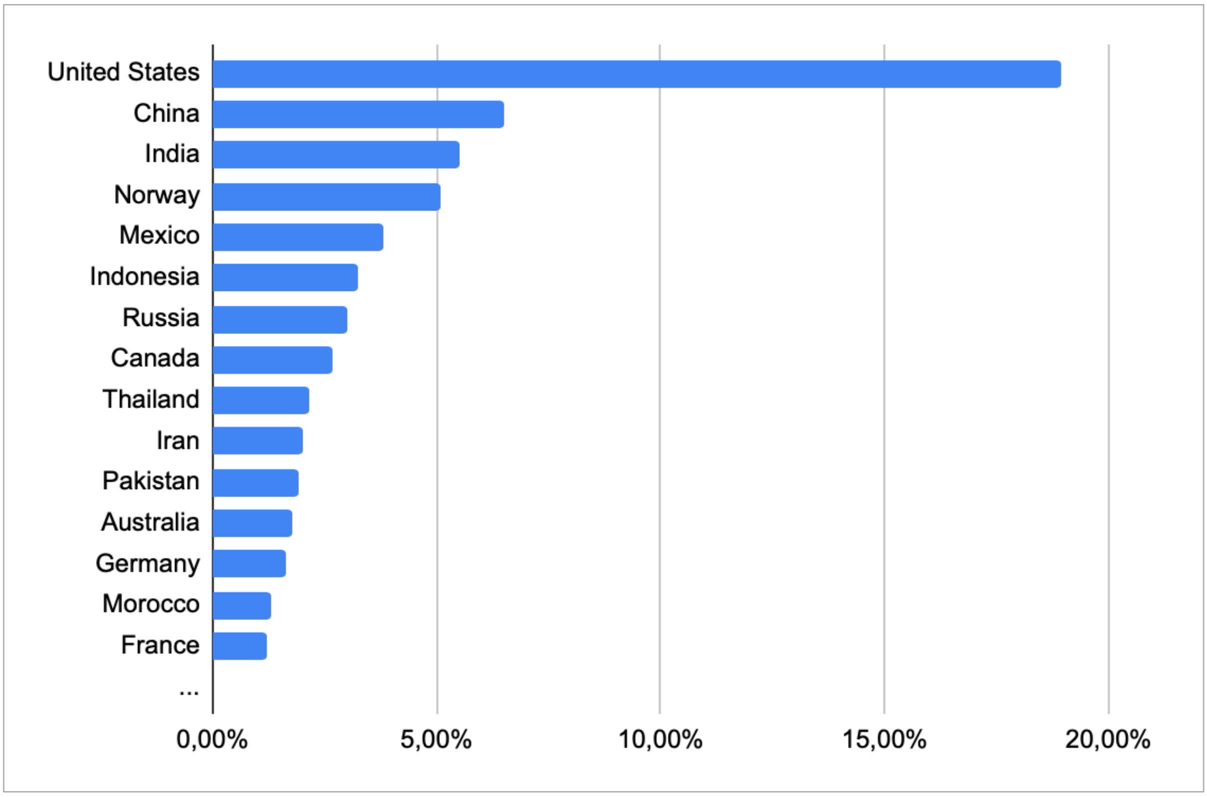
In contrast to GeoNames, a relational database whose entries form a large global hierarchy tree, Wikipedia can be understood as a loosely linked collection of knowledge in the form of semi-structured text. Its open format allows anyone to create articles about any topic or entity they consider relevant. Accordingly, Wikipedia has no fixed list of entity types, and its coverage cannot be measured in such terms. Instead, its coverage is only limited by relevance. For street-level geoparsing, this is at the same time an advantage and a disadvantage. If authors mention a location without further context, they assume that this location is known to the majority of their audience (compare Grice [1975]). This suggests that most locations referenced in news articles or works of literature should have their own Wikipedia article. The locations mentioned in the sample text about New York do for example all have long Wikipedia articles dedicated to them - even those not listed in GeoNames. At the same time, the majority of streets and facilities in cities and towns around the world do not have dedicated Wikipedia articles. In 2017 the DBpedia ontology contained ∼6.0 million places, of which ∼3.3 million were populated places[1]. These numbers serve as a rough estimate for the number of geo-annotated Wikipedia articles at that time. For comparison, GeoNames today contains ∼11.8 million places of which ∼4.7 million are populated places. This suggests that in total, Wikipedia covers even fewer street-level locations than GeoNames. It also shows that the ratio of street-level entities per populated place is very low for both ontologies[2].

For street-level geoparsing, however, this ratio should be as high as possible. And there are geographical ontologies that more than fulfill this requirement: map databases, like

[1] Data from `http://downloads.dbpedia.org/current/statistics/stats-types-stats.json` (accessed December 2019)

[2] This of course only considers the geo-annotated subset of Wikipedia. For toponym resolution, only this subset is relevant.





Data from `https://www.geonames.org/statistics/` as of January 2020

Figure 4.4: GeoNames feature class and country distribution

OpenStreetMap and Google Maps. These databases contain the exact geometries and names of streets, buildings and open spaces in populated places around the world. They therefore cover exactly those locations that this thesis intends to resolve. This qualifies map databases as ideal reference ontologies for street-level geoparsing. As gazetteers, however, map databases represent the extreme end of the precision-recall compromise, as they will produce a match for almost every term in the English language, and significantly more than one for most. But Middleton et al. [2013] show that local gazetteers derived from small map regions can be a valuable source of street-level toponyms. This suggests that narrowing down the search scope can make map databases feasible for toponym recognition and retrieval of resolution candidates.

4.3.3 Street-level Ambiguity

In the experiments above, many street-level entities were not recognized at all, and for those that were recognized the employed reference ontologies often were too limited. If a geoparsing system would have access to an ontology that covers the full spectrum of street-level entities (such as map databases) a new problem would arise: high referent ambiguity. Purves et al. [2018] show that street-level toponyms exhibit much higher levels of ambiguity than toponyms at region or city level. One reason for this is that there are far more street-level locations than geographical entities. The other reason is that the names of street-level entities only need to be unique within a small area (e.g. the populated place they are located in). Street-level entities also often repeat themselves in every populated place of the same cultural region, which leads to many instances with similar names (e.g. “Main Street”, “St. John’s Church”). The street names in the sample text about New York exemplify this high ambiguity. Wikipedia lists 50 different geo-annotated articles for the name “Broadway” [1] and OpenStreetMap contains a similar amount of elements with that name[2].

Resolving street-level toponyms is not only hard because of the number of candidates to disambiguate, but also because most traditional disambiguation heuristics do not work for them. For the default sense heuristic, the problem is that street-level entities lack a common property that could be used to compare their prominence. Even if two entities are of the same type, i.e. two streets, it is hard to find a universal rule set to determine which one of both is more relevant (is a longer or wider street more relevant? a highway or a pedestrian zone?). The ontological similarity heuristic cannot be applied because street-level toponyms are not part of the administrative hierarchy (although it is usually

[1] `https://en.wikipedia.org/wiki/Broadway` (accessed January 2019)

[2] Overpass query: `rel["name"="Broadway"]; out;` (loaded January 2019)

possible to relate them to a populated place). The sheer amount and the high granularity of street-level toponyms also make the training of geospatial language models infeasible. The only heuristic that can be used without adaptation is spatial minimality. This implies that in order to resolve ambiguous street-level toponyms, it is essential to determine the geographic context in which they are used.

4.3.4 Semantic Understanding

No matter how sophisticated the disambiguation heuristics or how comprehensive the gazetteer, as long as geoparsers do not understand the semantic relationships of a document, there will always be cases in which toponyms will be resolved incorrectly. For instance, practically every modern geoparser would resolve the sentence “She migrated from Georgia to the U.S.” wrong. Although both toponyms can easily be identified as countries through gazetteer lookup, all resolution heuristics used today would prefer for the U.S. state Georgia over the country Georgia here (the U.S. state is larger in population and surface area). To choose the correct sense, the system would have to understand the concept of migration, which requires two arguments that are geographically distinct. To cover not just one but all cases where semantic relations should override heuristics, the system would need a coherent semantic understanding of language in general. Unfortunately, achieving such semantic understanding remains a hard problem for artificial intelligence [Yampolskiy, 2013; Lucy and Gauthier, 2017; Radford et al., 2018]. Therefore this thesis will not attempt to pursue semantic reasoning, but will rather focus on leveraging geographic knowledge to enable new and better disambiguation heuristics.

4.4 Summary

This chapter first defined which functionality this thesis considers necessary for street-level geoparsing. It then proved that this functionality is not yet implemented by any of the tested state-of-the-art systems. An analysis of the observed limitations followed, which lead to a strategy to overcome these issues. The next chapter will present a new geoparsing system that leverages this strategy to globally recognize and resolve street-level toponyms.

Chapter 5

Method

This chapter proposes a new geoparsing technique, designed to recognize and resolve more street-level toponyms than currently available solutions. The chapter further presents the **txt2map** system, which implements this technique. The first section explains the hypothesis behind the technique. Then, the general approach and the individual steps of the technique are described. Finally, the txt2map system is introduced and its components documented.

5.1 Hypothesis

It was shown in Section 4.3.3 that constructing a single global street-level gazetteer - containing all names of all street-level entities of every populated place around the world - is not a feasible solution for street-level geoparsing. It was also argued that for toponym recognition, it is essential to know which names have a geographical meaning in order to decide which recognized named entities should be included in the resolution step.

Thus, a street-level geoparser must know which locations to look for, but can not know all locations. Notably, the same is true for human geoparsers, i.e. readers. No human being knows all locations listed in GeoNames (or OpenStreetMap), but most are still perfectly able to resolve the toponyms in the texts they read. This is because authors and readers share a common toponym vocabulary. If a geoparsing system can infer which toponyms the documents target audience can be expected to know, it should therefore be able to resolve most local references. Lieberman et al. [2010] were among the first to build a geoparsing system based on this hypothesis. As described in Section 2.2.1, they assume that there is a global lexicon of toponyms which are understood universally and a local lexicon that is shared by speakers of a specific region. They show that such local lexicons can be automatically inferred from geo-annotated documents that make use of the local vocabulary. Leveraging the same principle, Middleton et al. [2013] extend their gazetteers with names from OpenStreetMap to detect local references in tweets from specific regions. The caveat of both systems is that they require a priori knowledge about the geographic scope of the processed documents. The authors first select a region, then infer a local lexicon for that region, and then use that lexicon to parse documents targeted to audiences in that region.

The idea behind the approach presented here is the following: If it is possible to infer the local lexicon from a collection of text documents, then the obtained information must be present in the texts themselves. Conversely, a single text document can already contain information about the geographic scope of its target audience. In practice, this is often obvious to a human reader. If the title of an article contains the word "Dublin", then we assume that the pub mentioned in that article is located in Dublin, even if we never heard its name and know nothing about it. Formally, this can be explained by the overlap of the global and local lexicons described by Lieberman et al. [2010]. To a reader who does not share the local lexicon of a presented text, only the global toponyms are known. But the subset of global toponyms that is being used can hint at the geographical scope of the document. The approach presented here attempts to find exactly those "overlap" toponyms and use them to construct local gazetteers for street-level geoparsing. The assumption behind this is that it is possible to infer the local lexicon of a document's target audience from the geopolitical entities that it references.

The next section will describe practical approaches to extract such references and construct local gazetteers for them.

5.2 Approach

The geoparsing technique introduced above consists of three phases. The first phase involves extracting globally significant locations from the document text. Section 5.2.1 will introduce the term *local anchor point* for such locations and define which locations are meant by it. Section 5.2.2 then describes how such anchor points can be found. In the second phase, the local gazetteers that capture the geographic scope of the document are generated. This process is described in Section 5.2.3. In the last phase, the document is scanned for toponyms from those local gazetteers. This phase is discussed in Section 5.2.4.

5.2.1 Local Anchor Points

A local toponym lexicon is an abstract concept. To be able to work with it, some generalization will be made. First, it will be assumed that the target region of an article can be described as a radius around a coordinate. The center coordinates of such regions will be referred to as *local anchor points* in the following. Obviously, many documents do not have a geographically definable target audience, and many others target everyone on Earth. If local toponyms are used, however, it can be assumed that there is a regional audience, because only readers familiar with the region will know that the mentioned locations exist.

The next generalization is to equate target regions with cities and local anchor points with city centers. This is not always correct, because many authors write for larger administrative areas like counties or countries. For street-level toponyms, however, the assumption is again plausible, because such names are rarely known outside of the city they are located in[1]. On the other hand, knowing which city a street-level entity is located in is usually sufficient to disambiguate it. It should be noted that the term *city* is used here as a synonym for populated places, i.e. human settlements of all sizes. All such places represent a locally concentrated group of people that share a local toponym lexicon.

The circular shape of the constructed target regions is an approximation as well. In reality, target regions have soft borders which may or may not follow areas of settlement, or no borders at all. The rationale behind the radial shape is that the relevance of a toponym decreases with distance to the city center. If the target audience of a document are all people in a city, then locations in the center will be known by most readers, while locations in the suburbs will only be known by a small subset.

In conclusion, local anchor points are city center coordinates used to model the geographical scope of a document. In the ideal case, an article mentions one such anchor, which then yields one lexicon that fits the document scope. In practice, however, documents often contain the names of multiple cities. In such cases, it is not clear which city actually represents the document scope (if any). The strategy that will be pursued in this case is to treat all found anchors as possible scopes and use all derived lexicons for local matching. Another possibility is that a document uses local toponyms without mentioning the city they are located in. In these cases, both human readers and geoparsers would need additional contextual information to interpret local references.

The next sections will describe how local anchor points can be found, converted to local lexicons and utilized for street-level geoparsing.

[1] This is true for the vast majority of street-level toponyms. Of course, some names of tourist attractions are known nation- or world-wide.

5.2.2 Global Geoparsing

The goal of this phase is to scan the text for mentions of populated places that can serve as local anchor points. This is a classical geoparsing problem. Since the locations that are to be extracted here are still on the global level (i.e. have a population), established geoparsing methods can be applied. The main design decisions for geoparser are (1) whether to use a general-purpose NER tool for toponym recognition or train a custom model; (2) how to retrieve resolution candidates; and (3) which heuristics to use for disambiguation. These questions will be discussed in the following.

Although good results have been achieved with self-trained CRF models Inkpen et al. [2015]; Gritta et al. [2018a], many recently published geoparsing systems make use of pre-trained general-purpose systems [Berico Technologies, 2016; Kamalloo and Rafiei, 2018; Karimzadeh et al., 2019]. Using such a system has several advantages when searching for local anchor points. First, they are pre-trained on large linguistic corpora. Second, they can achieve high precision out-of-the-box. Third, they tend to classify street-level toponyms as organization or faculty entities, which has been listed as disadvantage before, but is actually beneficial for the use case described here. And fourth, they can be easily compared to choose the best performing option, as done by Karimzadeh et al. [2019]. The drawback is that such tools tend to sacrifice recall for precision.

This is problematic, because in the global recognition phase, recall is more important than precision. High recall is important because every false negative is a potentially missed local anchor point and can thus prevent the recognition of a whole set of street-level entities that would have been listed in a local gazetteer constructed from that anchor. A false positive, on the other hand, can only indirectly interfere with street-level recognition, either by biasing disambiguation heuristics to choose a wrong sense for another toponym, or by causing a false match on street level through the incorrectly generated local gazetteer. Both cases are less likely than the missed local anchor case as they represent a double coincidence.

The generally lower recall of general-purpose NER tools can be partially compensated by using a reclassification gazetteer, as described in Section 4.3.1. Such a gazetteer is used to catch toponyms that were incorrectly classified as other entity types by the NER tool. Here, the gazetteer should ensure that names of populated places, especially larger ones, are not misclassified as organizations or persons. With the addition of such a reclassification mechanism, the NER approach seems sufficient to recognize local anchor points. The decision for question (1) is therefore to use general-purpose solutions. The experiments in Chapter 6 will show how well these solutions really perform in the given scenario.

To obtain default and alternative location senses for the recognized toponyms a geographical database is required. In geoparsing, GeoNames is a default choice for this. In Section 4.3.2 it was further shown that GeoNames covers 4.7 million populated places worldwide, which is sufficient for local anchor extraction. Using GeoNames as reference ontology also simplifies evaluation, because many geoparsing corpora (GeoWebNews, LGL and others) link annotated toponyms to GeoNames entries. GeoNames is therefore the clear answer to question (2).

The selection of disambiguation heuristics for toponym resolution is the topic of whole publications (compare Section 2.2.3). The two heuristics that literature seems to unequivocally agree on are one-sense-per-referent and default sense, because the most populous candidate is often the already the correct one [Gritta et al., 2018a]. The challenge is to determine in which cases the most populous location is the wrong choice, and then to select an alternative. The commonly used heuristics to approach this task - spatial minimality, ontological similarity and language models - all seek disambiguation hints in the words that surround the toponym. The best results would therefore most likely be achieved with a holistic language model that takes into account the geospatial bias of each word in a context window. Such a model would have to be trained on a large corpus to accurately learn the geospatial distribution profiles of all words (compare [Gritta et al., 2018b]). But the purpose of this thesis is not to find new or better ways to leverage the available lexical evidence. Rather, the strategy for this step is to choose a proven method that is able to correctly resolve most anchor points. The heuristic that is considered most useful for this purpose is ontological similarity. Compared against spatial minimality it has the advantage that it can exploit the *contains* relation between cities and larger-scale entities, which allows for example to cluster Los Angeles and New York as U.S. American cities although they are almost 4000km apart. Using the administrative hierarchy as reference further makes it possible to prefer city level entities over their parents with the same name (i.e. chose the city of Québec instead of the much larger and more populous province). A reason not to use language model heuristics is also their low accuracy. A global grid tile of 100x100km is less useful for local gazetteer creation than a single coordinate that denotes the center of a populated place. For these reasons, the answer to question (3) will be to use the proven combination of default sense and ontological similarity heuristics, while sticking to the one-sense-per-referent principle (as suggested for example by Leidner [2007]).

In summary, the full global geoparsing phase consists of the following steps.

1. Recognize location entities using a NER tool
2. Attempt to reclassify using a gazetteer
3. Obtain location senses for all recognized toponyms from GeoNames
4. Resolve each recognized toponyms to its default sense

5. Sort the default senses entities into an administrative hierarchy tree
6. Find non-default senses with higher overall ontological similarity

Practical consideration for the implementation of each of these steps will be provided in Section 5.3. The result of these steps is a hierarchical tree of location entities, with each node connected to one or more toponyms in the text. From this tree, local anchor points can be obtained simply by selecting all nodes that are a certain distance away from the root node (i.e. the hierarchy layer for cities and all layers below). In some cases, it might also make sense to use entities from higher layers as local anchor points, for example if they represent a small geographic area (e.g. the Vatican). It is also sometimes possible to move down the administrative hierarchy to find city-level locations for recognized toponyms that would otherwise have no anchor (e.g. Québec city for Québec state).

The next section describes how the extracted local anchor points can be used to construct local gazetteers.

5.2.3 Local Gazetteer Construction

Local gazetteers should represent the toponym vocabulary of a specific region. The local anchor points from the previous phase represent the centers of such regions. The goal of this phase is to retrieve street-level entities for these regions and assemble them into gazetteers.

In the discussion about ontology coverage in Chapter 4 it was shown that map databases are a comprehensive source of street-level location data. A database that was used for this purpose before is OpenStreetMap [Middleton et al., 2013; Al-Olimat et al., 2018]. As many routing applications[1] use OpenStreetMap as data source, it can be also assumed that the data is up-to-date and accurate. It will therefore be the map database of choice for the approach described here.

OpenStreetMap offers a free data retrieval API called Overpass[2]. This API can be used to download elements within a specified region. It is further possible to filter elements by metadata, which allows to only include elements that have a name. The size of the loaded region can either be fixed (e.g. 15km around the anchor point) or dynamically calculated based on population or spatial extend of the geopolitical entity used as anchor. Section 5.3 will discuss techniques to filter the amount of retrieved data.

With the retrieved data, it is then possible to construct gazetteer data structures that

[1] Most notably GraphHopper `https://www.graphhopper.com`

[2] `https://wiki.openstreetmap.org/wiki/Overpass_API`

allow efficient lookup of toponyms and resolution of toponyms to OpenStreetMap entities. The next section describes how these gazetteers can be used to recognize and resolve street-level toponyms.

5.2.4 Local Geoparsing

Street-level toponyms are highly ambiguous on global scale. But within the city or town they are located in, they are usually unique. This means that the names in the local gazetteers from the previous phase should already be fully disambiguated. What remains to do is to scan the document for occurrences of names listed in the gazetteers. Two kinds of ambiguity can still occur in this process. First, street-level toponyms can also be homonyms or metonyms, i.e. they can look like a toponym listed in the local gazetteers, but refer to a non-locational entity. No satisfying solution was found for this problem. The second possibility is that a toponym is listed in two local gazetteers for different regions. A practical solution for such cases is to resolve to the entry from the gazetteer of the most populous anchor point. This leads to the following local geoparsing strategy:

1. Find all occurrences of toponyms listed in the local gazetteers
2. Filter out occurrences that are already annotated otherwise
3. Resolve the rest according to the local gazetteer they are listed in
    (a) If a toponym occurs in multiple gazetteers, prefer the largest anchor

To account for inconsistencies in the OpenStreetMap data (which is crowd-sourced) fuzzy matching can be considered in step 1. Enriching gazetteers with abbreviated variants of toponym (e.g. replacing “Street” with “St” and “Avenue” with “Ave”) has also been shown to increase recall [Middleton et al., 2013]. Both methods seem promising, but will not be further discussed here as they go beyond the scope of this thesis (see Chapter 1).

A difficult decision is to determine whether a local match should overwrite an annotation from the NER tool. One strategy is to always overwrite previous annotations, as the NER tool has no document-specific knowledge while the derived gazetteer should match the document scope. However, unfiltered OpenStreetMap data also has the potential to produce many false positives. Depending on the chosen distance around the local anchor point and the population density at that point, a local gazetteer can contain several 10,000 toponyms. The strategy chosen here values the machine-learned classification produced by the NER higher than the rule-derived local context. However, it should be possible to craft a set of rules or train a model that can make an informed decision at this point.

Finding local gazetteer matches is the final phase of the three-phase street-level geoparsing approach presented here. The end result is a document annotated with references to geographical ontologies, which cover the full spectrum of named locations on Earth, from continents to bus stops. The next section will present a concrete geoparsing system that implements this three-step process.

5.3 The txt2map System

The section introduces the **txt2map** open-source geoparsing system[1]. It realizes the street-level geoparsing technique described above. The first part presents the high-level architecture of the system. Then, the individual modules will be described in detail. The main focus hereby lies on the documentation of technical decisions that might also be relevant for other systems. Each section will further point out parameters that have a direct impact on the system's overall performance and can be tuned to adapt the approach to new use cases.

5.3.1 System Architecture

The txt2map system is structured as a text processing pipeline. The central data structure is a document, which is transferred from one processing step to the next. Each processing step adds an additional layer of annotations. The core system consists of four text processing modules. Figure 5.1 shows how these modules are arranged in a pipeline.

The first module performs Named Entity Recognition. The actual labeling task is delegated to a stand-alone NER. This allows the architecture to support multiple NER tools in parallel. The NER module sends the text to the NER server, receives the results and annotates the document accordingly.

The second module performs entity reclassification. To do so it matches names annotated as persons or organizations against a local gazetteer of significant city names. The module creates a new annotation layer in which the entity class labels are adjusted.

The third module is the global geoparser. Its main purpose is to extract local anchor points. To do so, it resolves the recognized toponyms to GeoNames entries and arranges them in a hierarchical tree. Resolution candidates are retrieved from the GeoNames API. It adds two annotation layers, one that associates recognized toponyms with GeoNames identifiers, and one that marks the toponyms that should serve as local anchor points.

The fourth and last module is the local geoparser. It combines the tasks of local gazetteer construction and local geoparsing. To construct local gazetteers it takes the local anchor

[1] Available at `https://github.com/ernesto-elsaesser/txt2map`

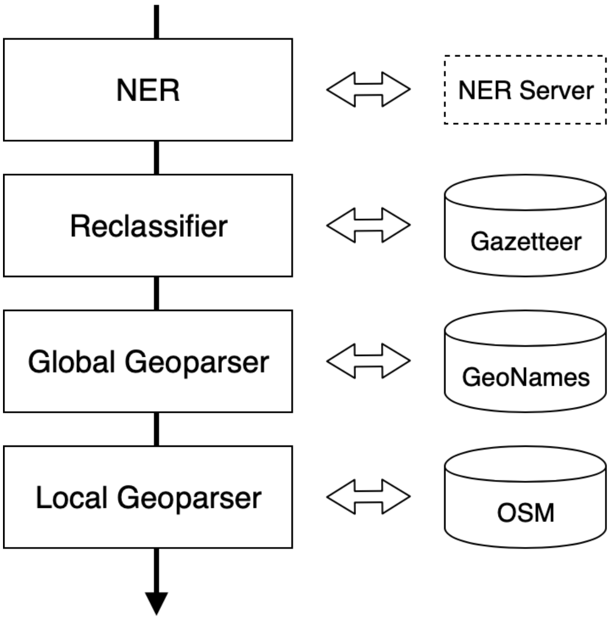


Figure 5.1: The txt2map processing pipeline

points from the respective annotation layer, clusters them to avoid redundant loading and then retrieves gazetteer data for OpenStreetMap. Then the module scans the document for mentions of the received toponyms and decides whether the mentions should be annotated. The annotations are then written into a new annotation layer.

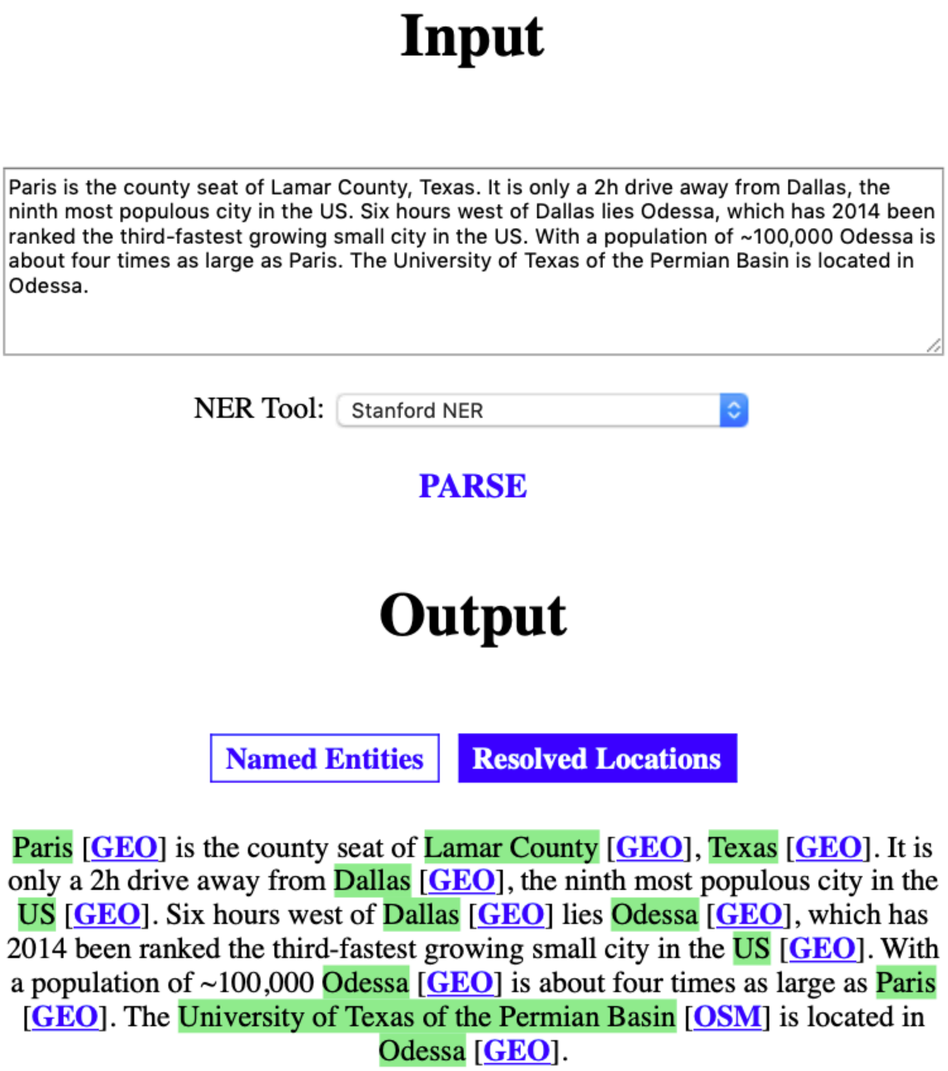
The input of the pipeline is an unannotated document and the output is the same document with five layers of annotations. The published system further includes a web application that configures and executes the pipeline for the user and visualizes the produced annotation layers. Figure 5.2 shows a screenshot of the web interface.

5.3.2 Named Entity Recognition

The NER module serves as an adapter between the txt2map system and third-party NER tools. At the time of publication, the module supports four NER tagger:

- spaCy
- The Illinois NE Tagger (part of the CogComp framework)
- Stanford NER
- The Google Cloud Natural Language API

For the former three, the software package contains server implementations that provide an HTTP endpoint for recognition. The latter is a hosted service. All four systems have been described in Chapter 3. They all produce entity labels but use different classification



Paris is the county seat of Lamar County, Texas. It is only a 2h drive away from Dallas, the ninth most populous city in the US. Six hours west of Dallas lies Odessa, which has 2014 been ranked the third-fastest growing small city in the US. With a population of ~100,000 Odessa is about four times as large as Paris. The University of Texas of the Permian Basin is located in Odessa.

NER Tool: Stanford NER

Paris [GEO] is the county seat of Lamar County [GEO], Texas [GEO]. It is only a 2h drive away from Dallas [GEO], the ninth most populous city in the US [GEO]. Six hours west of Dallas [GEO] lies Odessa [GEO], which has 2014 been ranked the third-fastest growing small city in the US [GEO]. With a population of ~100,000 Odessa [GEO] is about four times as large as Paris [GEO]. The University of Texas of the Permian Basin [OSM] is located in Odessa [GEO].

Figure 5.2: The txt2map web interface

schemes. The NER module contains client implementations for all four web services and converts their output to the internal annotation scheme used by txt2map (which only uses location, organization and person labels). The Google Cloud Natural Language API (GCNL) does not just recognize entities but also links them to Wikipedia articles (NER + EL). If needed, the NER module can convert those Wikipedia references into an additional annotation layer.

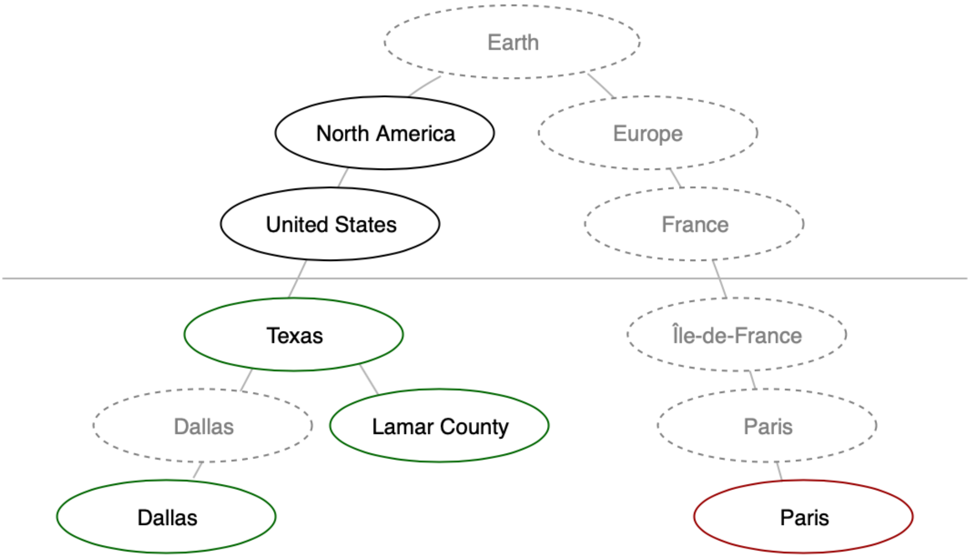
5.3.3 Reclassification

The reclassification module was introduced to increase toponym recognition recall. It uses a local gazetteer extracted from GeoNames. The gazetteer contains all entries with a population above 50,000. This value was chosen to include as many relevant local anchor points as possible while keeping the false positive rate at an acceptable level. The lower the population limit, the more ambiguous and homonymous names will be included in the gazetteer. The reclassifier takes all names labeled as organizations by the NER tool and checks if they are listed in the gazetteer. If a name is listed, its entity type is changed to location.

5.3.4 Global Geoparsing

The global geoparsing module performs toponym resolution. It is modeled after the systems published by Kamalloo and Rafiei [2018] and Karimzadeh et al. [2019]. The main difference to those systems is that the module does not use an offline search engine to retrieve GeoNames entries, but simply uses their public geocoding API. This API ranks results by relevance, so it already provides a default sense for each toponym. The results are being filtered by GeoNames feature class. Three classes are excluded from resolution: spot features (“S”), road and railway features (“R”) and undersea features (“U”). Spots, roads and railways are excluded because they represent street-level entities and the goal of the global geoparser is to extract references to geopolitical entities. Because spot features make up ∼20% of all GeoNames data, excluding them also simplifies the disambiguation task.

To also support the resolution of non-default sense, the ontological similarity heuristic is used. This is implemented as graph-based clustering. First, all recognized toponyms are assigned to their default sense. These obtained locations are then sorted into a tree structure that represents the administrative hierarchy. All nodes below layer three of this hierarchy are then treated as one cluster (state/province level). Now the algorithm iterates over all clusters that contain only one location entity and no direct parents. These entities are marked as unsupported. In the example in Figure 5.3, the toponym “Paris”



green - supported; red - unsupported; dotted - not present

Figure 5.3: Sample hierarchy tree during the disambiguation process

was resolved to its default sense in France. As the text does not mention any other entity in this segment of the tree, the entity is flagged as unsupported. The cluster in Texas meanwhile supports itself, and is additionally supported by the common ancestor "United States". For toponyms flagged as unsupported, alternative senses are considered. Going from largest to smallest, each alternative sense is sorted into the hierarchy tree[1]. If the alternative sense is supported by the tree, it is kept. Otherwise, it is removed and the next sense is tried. If no supported alternative is found, the default sense is kept. This algorithm was chosen to ensure that alternative senses are only selected if they are supported by other default sense resolutions and only one at a time. An alternative implementation that updated the complete resolution set at once produces more incorrect resolutions. Because the default sense correctly resolves 80-90% of all toponyms already (compare for example [Gritta et al., 2018a]), swapping senses should be the exception, not the norm.

The ontological similarity heuristic iterates over all clusters until no more supported alternative senses can be found or no more unsupported clusters are left. All leaves that are below the clustering line (layer four and deeper) after the algorithm finished are used as local anchor points.

[1] In practice, the algorithm stops after 10 candidates for each toponym. The GeoNames API returns several 100 entries for some names.

5.3.5 Local Geoparsing

The local geoparsing module is the one that differentiates the txt2map system from other geoparsers. It implements both the local gazetteer construction and the local geoparsing routine. Its output is an additional layer of annotations that includes street-level references not recognized or resolved by previous steps.

The first task of the local geoparser is to execute the Overpass queries that load the required gazetteer data from OpenStreetMap. The module uses the same Overpass QL[1] template for every local anchor point. That template requests all OpenStreetMap elements within a 15km bounding box around the anchor coordinate, with some restrictions. The first restriction is that the element must have a name, because otherwise it could not be mentioned in text anyways. The other restrictions exclude certain groups of elements that often have names but will rarely be mentioned in texts targeted at larger audiences. These include office buildings and shops as well as electrical power lines and the associated infrastructure[2]. Optionally, restaurants can be excluded as well. To reduce the amount of data that needs to be transferred, only properties that contain name information are retrieved. Loading times can be further optimized by reducing the size of the requested bounding box, but this can impede geoparsing performance.

The received data is stored locally in SQLite databases. This is mainly a caching mechanism to avoid repeated loads of large gazetteers. The databases vary in size, from only several KB for rural, sparsely populated locations, to over 15 MB for metropolitan cluster areas. Such databases can contain over 100,000 unique names. Obviously, loading such amounts of data can lead to noticeable delays for the users of the system when a local anchor is processed for the first time.

When OpenStreetMap data is available, the local geoparsing process begins. The process follows the three steps listed in Section 5.2.4. Simply put, the algorithm scans the document for each of the names obtained from OpenStreetMap and annotates each match that is not already annotated. The matching algorithm is implemented efficiently so that gazetteer size has no impact on performance. For every text token that starts with an upper case character or a number, a prefix search is performed to retrieve possible completions from the gazetteers. The algorithm then chooses the longest completion that matches the text. By starting with the gazetteer of the largest anchor point (in terms of population), the matching algorithm ensures that for duplicate matches the OpenStreetMap reference from the most prominent geographic context is chosen. To avoid

[1] Overpass QL is the query language for the Overpass API.

[2] OpenStreetMap covers electrical infrastructure is minute detail. `https://wiki.openstreetmap.org/wiki/Key:power`

false positives, only matches longer than four characters are annotated.

The local gazetteer matching represents the last step in the txt2map pipeline. The produced annotations can be used to render a local map or index documents for location-aware content discovery. Through the linked GeoNames and OpenStreetMap entities, rich metadata is available for each reference. As most GeoNames entries are linked to Wikipedia pages, it is for example possible to use txt2map for "Wikification" [Mihalcea and Csomai, 2007]. Some OpenStreetMap elements are associated with Wikipedia pages as well[1].

The next chapter will evaluate the performance of the txt2map system and compare it to other geoparsing systems. It will also be measured which of the four supported NER tools works best in conjunction with the other pipeline steps.

[1] `https://wiki.openstreetmap.org/wiki/Key:wikipedia`

Chapter 6

Evaluation

This chapter presents evaluation results for the txt2map geoparser and compares its performance with that of other state-of-the-art systems. It will be shown that the generated gazetteer approach has advantages over other geoparsing systems on the street level. This chapter will first introduce the corpora and metrics used for evaluation and then present experimental results to support that statement. It closes with a critical interpretation of the presented results.

6.1 Corpora

Gritta et al. [2018a] recently published a comprehensive report about available resources and methods for geoparsing evaluation. For the comparison of geoparsing performance, the report recommends the WikToR, LGL, GeoVirus, TR-News and GeoWebNews corpora (which also seem to be the only publicly available geoparsing corpora in existence). Among these, **LGL** and **GeoWebNews** are the only ones that include annotations for street-level toponyms. LGL was created by Lieberman et al. [2010] to validate their local lexicon theory and therefore contains many local toponyms. GeoWebNews, on the other hand, was designed with the general toponym resolution task in mind and is supposed to enable a fair comparison between different systems [Gritta et al., 2018a]. Together those two corpora serve as a good benchmark for both overall and street-level geoparsing performance. While not representative in any form, the sample texts from Chapter 4 will also be included in the evaluation, mainly to demonstrate how general-purpose NER, reclassification gazetteer, ontological similarity heuristic and local gazetteer inference play together in individual cases.

6.2 Metrics

The two steps involved in geoparsing, toponym recognition and toponym resolution, are structurally different. The task of toponym recognition is to assign labels to sequences while the task of toponym resolution is to assign coordinates to labels. Because of this, it is common to evaluate both steps separately using different metrics [Leidner, 2007; Gritta et al., 2018a].

Toponym recognition is a sequence labeling task, which is usually modeled and evaluated as a classification problem. Accordingly, the standard metrics for toponym recognition are precision, recall and F1-score. This thesis will stick to this convention.

Toponym resolution produces coordinates, which are (unlike labels) not just right or wrong, but differ from the gold coordinate by a measurable distance. Applying classification metrics that count true positives, true negatives, false positives and false negatives would therefore require a conversion rule[1]. This conversion rule would have to be considered when interpreting and comparing results. In order to standardize geoparsing evaluation, Gritta et al. [2018a] therefore recommend an alternative set of unambiguous metrics that leverage the error distance information. In my opinion the most intuitive and comprehensible of those metrics is Accuracy@Xkm. It is used by several authors explicitly [Speriosu and Baldridge, 2013; DeLozier et al., 2015; Santos et al., 2015] or implicitly when reporting precision [Kamalloo and Rafiei, 2018; Lieberman and Samet, 2012]. Accuracy@Xkm measures the percentage of annotated toponyms that have been resolved to coordinates within an X kilometer radius of the gold coordinate. A common choice for X is 161 kilometers (10 miles), as this is enough to compensate for varying center coordinates of geopolitical entities among different ontologies or between different hierarchy levels [Lieberman et al., 2010; Lieberman and Samet, 2012; Kamalloo and Rafiei, 2018].

A characteristic of Accuracy@Xkm is that it treats all locations outside of the X kilometer circle around the target location as wrong resolutions, no matter if they are just outside that radius or on the other side of the planet. For street-level geoparsing this is appropriate, as it makes no difference to the user whether a toponym was resolved to a nearby city or another continent if it does not appear in the local scope that he or she is interested in.

[1] Some recent systems use statistical classifiers for toponym resolution. To transform the task into a classification problem they divide the globe into equally-sized tiles that serve as classes (see Section 3.2). For these systems classification metrics are obviously the natural choice.

6.3 Evaluated Configurations

Due to the modular architecture of txt2map it is possible to use different NER system in the initial toponym recognition step. For evaluation, three different NER annotators were implemented:

- **spaCy NER** - large model, GPE and LOC entities
- **Stanford NER** - CoNLL model, LOCATION entities
- **The Illinois NE Tagger ("CogComp")** - CoNLL model, LOC entities
- **Google Cloud Natural Language API ("GCNL")** - LOCATION entities

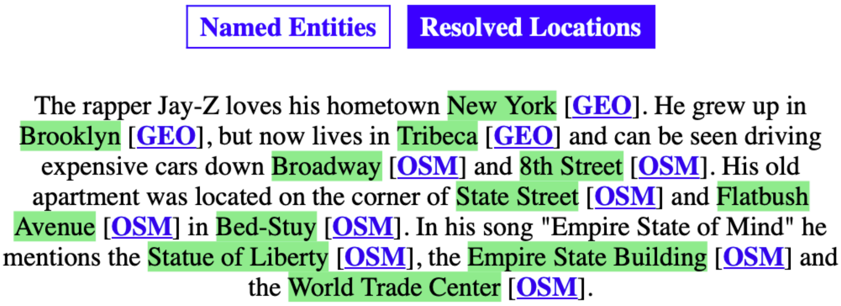
Google Cloud Natural Language was chosen because it performed best in the initial tests in Chapter 4 and is further the best publicly available systems from the ones evaluated by Gritta et al. [2018a]. Illinois NET was chosen because it performed best in the experiments conducted by Karimzadeh et al. [2019] and produced good results for the sample texts presented earlier. Stanford NER was chosen because it is a common choice for toponym recognition in several geoparsing systems (see Chapter 3) and performed well in both experiments just mentioned. spaCy was chosen although its street-level recognition performance in initial tests was poor. Above street level, however, it will be shown that spaCy's toponym recognition capabilities are on par with that of the other systems. Because of the local gazetteer approach, the omission of street-level toponyms in the global recognition phase is not an issue and can actually improve resolution results. It is far more important for the chosen NER tool to recognize the names of geopolitical entities. The results presented below suggest that spaCy does this reasonably well. The chosen models and entity types are the ones that produced the best results in initial tests (compare Table 4.3). The results of these tests mirrored the initial assumptions that larger models and models with fewer entity types would perform better (as stated in Section 4.3.1).

Because of its good performance in the experiments in Chapter 4, the Google Cloud Natural Language API is further included as a reference system for toponym resolution. The Wikipedia API[1] is used to retrieve coordinates for the linked Wikipedia articles.

To represent the current state-of-the-art in heuristics-based geoparsing, the TopoResolver system presented by Kamalloo and Rafiei [2018] will also be included in the measurements. The system was used in its standard configuration, as published on GitHub[2]. This configuration uses the "Context Hierarchy Fusion" approach which performed best for most corpora in the experiments conducted by Kamalloo et. al.

[1] `https://en.wikipedia.org/w/api.php`

[2] `https://github.com/ehsk/CHF-TopoResolver`



The rapper Jay-Z loves his hometown New York [GEO]. He grew up in Brooklyn [GEO], but now lives in Tribeca [GEO] and can be seen driving expensive cars down Broadway [OSM] and 8th Street [OSM]. His old apartment was located on the corner of State Street [OSM] and Flatbush Avenue [OSM] in Bed-Stuy [OSM]. In his song "Empire State of Mind" he mentions the Statue of Liberty [OSM], the Empire State Building [OSM] and the World Trade Center [OSM].

Figure 6.1: Sample output from the txt2map web interface

6.4 Results

This section presents experimental results for the systems and corpora mentioned above.

6.4.1 Sample Texts

txt2map is able to correctly recognize and resolve all toponyms in the two sample texts used to test geoparsing performance. For both texts, the results do not depend on the selected NER tool.

For the sample text about New York (Figure 4.1) the reclassification gazetteer will recognize the local anchor points “New York” and “Brooklyn” in any case. The OpenStreetMap gazetteers generated for those anchor points cover all locations mentioned in the text. Even “Bed-Stuy” as short name for Bedford-Stuyvesant is listed. The GeoNames API correctly geocodes the toponym as well. “8th Street” can be resolved because the local gazetteer matching algorithm also considers tokens that start with numbers. Figure 6.1 shows the output on the txt2map web interface.

For the sample text about ambiguously named cities in Texas (Figure 4.3) all toponyms except “Lamar County” are covered by the reclassification gazetteer. “Lamar County” is correctly recognized by all NER tools. The ontological similarity heuristic further correctly selects the non-default senses in Texas for the toponyms “Paris” and “Odessa”.

As demonstrated in Chapter 4, neither the Google Cloud Natural Language API nor the TopoResolver system can fully resolve those two texts.

6.4.2 GeoWebNews

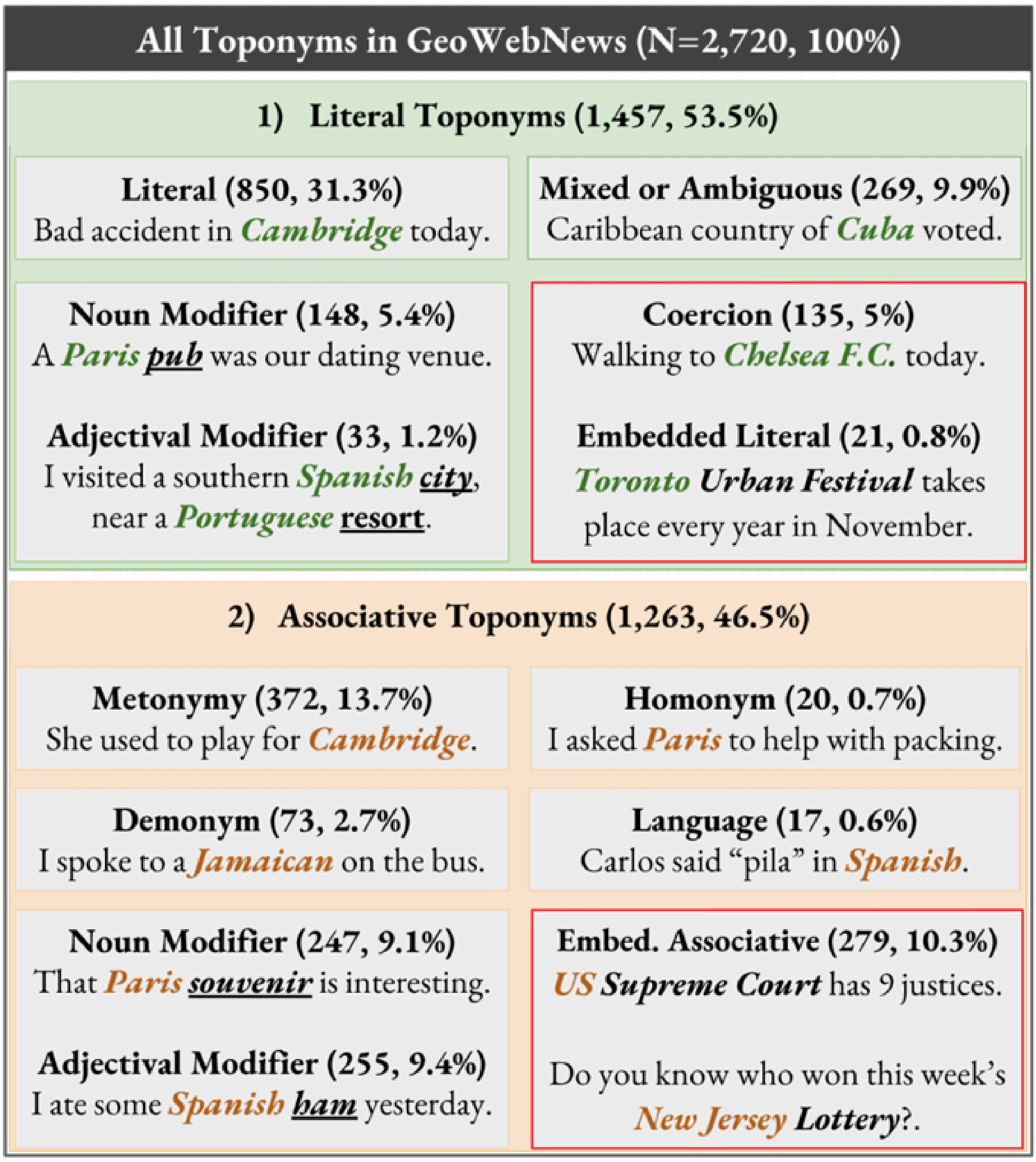
GeoWebNews is a manually annotated dataset of 200 articles from globally distributed newspapers (see Section 2.2.4). The corpus includes 2,720 annotated toponyms which are either linked to GeoNames identifiers or coordinate pairs. Each toponym is further categorized according to the toponym taxonomy presented by Gritta et al. [2018a] (see Figure 6.2). For the purpose of this paper, not all annotated toponym categories are relevant. Some do actually impair comparability, because different evaluated systems handle them differently. For this reason, the measurements here only consider the following toponym categories: (1) "Literal" which include all toponyms that directly refer to a physical location (i.e. the classical definition of a toponym); (2) "Mixed" which include toponyms that can be interpreted in a literal sense but also for example as political entity or a groups of people; (3) "(Literal) Noun Modifier" which include toponyms that precede or succeed a noun to indicate that that noun is either located within the referred entity; and (4) "Metonym". (2), (3) and (4) are included because they are syntactically not distinguishable from literal toponyms, and the semantic distinction is beyond the capabilities currently available NER tools. They also all represent useful local anchor points for gazetteer inference. For (2) the authors themselves recommend to include those toponyms in the literal group. Hence all four mentioned categories will be treated as proper, location-representing toponyms here. Excluded are all categories whose spelling differs from the canonical form (adjectival modifiers, demonyms, languages) as well as homonyms and embedded forms. The 1639 included toponyms represent approx. 60% of all annotations.

The annotated toponyms in the GeoWebNews corpus are resolved in two different ways. Most toponyms are annotated with GeoNames identifiers. Those toponyms that have are not covered by GeoNames are annotated with raw coordinates instead. The authors make the following statement about those coordinate-annotated toponyms:

> [S]ome toponyms (∼8%) require either an extra source of knowledge such as Google Maps API, a self-compiled list of businesses and organisations [sic] names [...] or even human-like inference to resolve correctly. These toponyms are facilities, buildings, street names, park names, festivals, universities and other venues. We have estimated the coordinates for these toponyms, which do not have an entry in Geonames using Google Maps API. These toponyms can be excluded from evaluation, which is what we did, due to the geoparsing difficulty. [Gritta et al., 2018a]

Notably, this exactly matches the definition of street-level toponyms used in this thesis. Therefore, this group will be included here, and will specifically serve as an indicator for street-level geoparsing performance. Approx. 7% of all "proper" toponyms defined above belong to this group.

The remainder of this section will present evaluation results for NER, toponym recognition and toponym resolution performance over the described subsets of the GeoWebNews corpus.



All Toponyms in GeoWebNews (N=2,720, 100%)

1) Literal Toponyms (1,457, 53.5%)

Literal (850, 31.3%)
Bad accident in *Cambridge* today.

Mixed or Ambiguous (269, 9.9%)
Caribbean country of *Cuba* voted.

Noun Modifier (148, 5.4%)
A *Paris* ***pub*** was our dating venue.

Adjectival Modifier (33, 1.2%)
I visited a southern *Spanish* ***city***, near a *Portuguese* **resort**.

Coercion (135, 5%)
Walking to *Chelsea F.C.* today.

Embedded Literal (21, 0.8%)
Toronto ***Urban Festival*** takes place every year in November.

2) Associative Toponyms (1,263, 46.5%)

Metonymy (372, 13.7%)
She used to play for *Cambridge*.

Homonym (20, 0.7%)
I asked *Paris* to help with packing.

Demonym (73, 2.7%)
I spoke to a *Jamaican* on the bus.

Language (17, 0.6%)
Carlos said "pila" in *Spanish*.

Noun Modifier (247, 9.1%)
That *Paris* ***souvenir*** is interesting.

Adjectival Modifier (255, 9.4%)
I ate some *Spanish* ***ham*** yesterday.

Embed. Associative (279, 10.3%)
US ***Supreme Court*** has 9 justices.

Do you know who won this week's *New Jersey* ***Lottery***?.

Figure 6.2: The "Pragmatic Taxonomy of Toponyms" presented by Gritta et al. [2018a] including corpus statistics for GeoWebNews

Recognition Method	P	R	F1
spaCy (NER only)	**0.76**	0.77	0.77
spaCy (reclassified)	0.75	0.78	0.77
CogComp (NER only)	**0.76**	0.75	0.76
CogComp (reclassified)	**0.76**	0.79	0.77
Stanford (NER only)	0.74	0.79	0.77
Stanford (reclassified)	0.75	0.82	0.78
GCNL (NER only)	**0.76**	0.91	**0.83**
GCNL (reclassified)	**0.76**	**0.92**	**0.83**

Table 6.1: Toponym recognition results for the GeoWebNews corpus

Recognition

To compare toponym resolution performance the corpus was annotated with all four NER tools, once only using the tool itself and once using the combination of NER tool and reclassification gazetteer. Measured were precision (P), recall (R), F1-score (F1) over the selected subset of the corpus. The Google Cloud Natural Language API does annotate demonyms as location entities. To achieve comparable precision scores, the demonyms were therefore filtered out in a post-processing step before the measurements. Table 6.1 shows the measured results.

The results show that reclassification only has a small impact on recall (1-3%). This can either mean that most names were already classified correctly, or that the reclassification algorithm can be optimized. An additional measurement was made to determine which is actually the case. In the measurement, the recall calculation was altered to treat non-location entities as true positives as well. Table 6.2 shows the obtained recall values when organization and/or person entities counted are counted as locations. The left column replicates the original values from Table 6.1. The numbers show that up to 11% of toponyms get misclassified as organizations or persons. Many of the toponyms misclassified as organizations refer to facilities, which are at the same time locations and organizations. In these cases, the NER classification is technically correct, but focuses on the wrong aspect of the entity for the purpose of geoparsing. This confirms that reclassification is a valid approach. Another interesting observation here is that the misclassification rate of GCNL is significantly lower than those of the other NER tools (3%), which explains its good overall performance.

NER Recall	LOC	LOC+ORG	LOC+ORG+PER
spaCy	0.77	0.83	0.86
CogComp	0.75	0.84	0.85
Stanford	0.79	0.87	0.90
GCNL	0.91	0.93	0.94
LOC - locations; ORG - organizations; PER - persons			

Table 6.2: NER recall for the GeoWebNews corpus including different entity classes

Resolution

Resolution performance was measured separately for the GeoNames and the coordinates group. For the GeoNames group Accuracy@161km ("A@161") was used and Accuracy@2km ("A@2") was used for the coordinates group. These tolerances allow some discrepancies between reference ontologies, but are still small enough to detect wrong resolutions. It is unlikely that two unique street-level entities with the same name exist within a two kilometer radius. A possible edge case for this metric is that the gold coordinate and the resolved coordinate of a long street are more than two kilometers apart. To account for this case, the tolerance distance was measured from the closest point on the outline of an entity's geometric shape. For both groups, the metrics are measured once for all recognizable toponyms and once for the subset of toponyms for which disambiguation candidates were found ("Res.").

GCNL resolves to Wikipedia URLs instead of coordinates. To calculate the error distances for the Accuracy@Xkm metrics, it is necessary to retrieve coordinates for each Wikipedia URL. This is done through the Wikipedia API mentioned above. If the API provides no coordinates for an article, the toponym is treated as not resolved.

The measured results are presented in Table 6.3. The "Global" section refers to the group annotated with GeoNames identifiers, the "Street Level" section to the group annotated with coordinates. The "Res. %" columns state how many of the annotated toponyms were resolved (correctly or incorrectly), providing context for the relative values in the column to their left.

Again, GCNL produces the best results. The highest resolution performance for both groups is achieved by txt2map using GCNL for NER. All txt2map pipelines perform better than the TopoResolver system from Kamalloo and Rafiei [2018]. As expected, its reliance on GeoNames prevents it from resolving most street-level references. The GCNL + Wikipedia solution produces remarkably good results for both groups. On street level, the combination actually outperforms the txt2map pipeline with spaCy as NER tool. The lower street-level performance for the spaCy pipeline is a result of spaCy's annotation

Resolution Method	Global			Street Level		
	A@161	A@161 Res.	Res. %	A@2	A@2 Res.	Res. %
txt2map (spaCy)	0.80	0.92	87%	0.21	0.43	50%
txt2map (CogComp)	0.78	0.92	85%	0.27	0.48	55%
txt2map (Stanford)	0.80	0.92	87%	0.27	0.48	56%
txt2map (GCNL)	**0.84**	0.91	**93%**	**0.28**	0.45	**63%**
GCNL + Wikipedia	0.67	0.82	82%	0.22	0.81	27%
TopoResolver	0.64	0.76	85%	0.03	0.19	14%

Table 6.3: Toponym resolution results for the GeoWebNews corpus

scheme. The framework follows the OntoNotes scheme and therefore uses a separate class for facilities. For the txt2map, it was decided not to consider this class for resolution, because it would lead to many incorrect resolutions because of coincidental GeoNames name matches. This increases resolution precision but also reduces the Accuracy@2km score reported here.

With ∼92% correct resolutions, the disambiguation algorithm based on ontological similarity seems to perform equally well for all NER tools. For reference, Gritta et al. [2018a] report that their state-of-the-art CNN-based toponym resolution system achieves 95% Accuracy@161km among all resolved toponyms from the GeoNames group (on a slightly different subset of the corpus). This suggests that more sophisticated disambiguation heuristics could further improve the overall performance of the txt2map architecture. For a system optimized for street-level resolution, 92% global accuracy is still a respectable result.

6.4.3 Local-Global Lexicon

The *Local-Global Lexicon* (LGL) is a manually annotated corpus of 588 news articles collected from 78 local newspapers (see Section 2.2.4). Of the 5088 annotated toponyms, 4462 are linked to GeoNames entities (∼88%). The remaining 626 toponyms, which are annotated but not resolved to geographical entities or coordinates, fall into the street-level category. As no coordinate data is available for those toponyms, this subset of the corpus cannot be used to measure resolution accuracy. What can be measured is the percentage of street-level toponyms resolved at all. This value is also indicative of geoparsing performance, because it expresses how much of the corpus' street-level vocabulary is known to the geoparser. Comparing the number of resolved toponyms can thus help to assess whether the dynamically generated gazetteers really make a difference. With over five times more annotated toponyms, the street-level subset of LGL is much

Resolution Method	Extracted Street-Level References
txt2map with local gazetteers	**61%**
txt2map without local gazetteers	**37%**
GCNL + Wikipedia	28%
TopoResolver	24%

Table 6.4: Local-Global Lexicon street-level toponym resolution results

larger than that of the GeoWebNews corpus, which only includes 113 toponyms.

LGL is also a more challenging corpus than GeoWebNews, because it was designed to contain many non-default-sense toponyms [Lieberman et al., 2010; Gritta et al., 2018a]. Many articles were taken from local newspapers which frequently report about cities that share their name with much larger ones elsewhere, like Paris, Texas or Alexandria, Virginia (both U.S.). Because authors and readers of such articles share a similar local lexicon, the authors can further use these non-default senses without providing additional disambiguation hints, which makes the geoparsing tasks for these articles even harder (if no metadata is used).

In contrast to GeoWebNews, LGL does not provide taxonomic information about the annotated toponyms. But as street-level toponyms are in almost all cases literal toponyms[1], this is not an issue here.

For the LGL measurements, only the best performing txt2map pipeline using GCNL was used. The measured results are presented in Table 6.4. It can be seen that the usage of local gazetteers significantly improves toponym coverage, achieving levels far above both the TopoResolver system and GCNL + Wikipedia.

6.5 Interpretation

The results presented in this chapter confirm several assumptions made throughout this thesis, but also reveal some potential for optimization.

The central hypothesis of the presented technique is that local lexicons can be inferred from globally significant location references. The data in Table 6.3 and Table 6.4 implies that the txt2map approach does indeed recognize more street-level toponyms than other

[1] Most of the categories that were excluded from GeoWebNews simply cannot be constructed with entities that are not associated with groups of people or cultures (demonyms, metonyms, adjectival modifiers, languages). Homonyms, noun modifiers and embedded forms are technically possible, but assumably very rare.

modern geoparsing and EL systems. The results suggest that at least for some documents, local lexicons are inferred correctly.

Another assumption of this thesis is that general-purpose NER tools tend to misclassify toponyms as other entity types. This recall values in Table 6.4 support this assumption. However, the reclassification gazetteer could only marginally increase recall. This component should therefore either be improved or replaced by more efficient solutions.

The main goal of the experiments presented here was to compare the performance of txt2map and the local gazetteer approach to the best available alternatives. The TopoResolver system was selected to represent the current state-of-the-art in rule-based toponym resolution. The systems presented by Karimzadeh et al. [2019] and Berico Technologies [2016] are expected to perform similarly. GCNL was selected because it performed even better than the tested systems from academia, and also better than other commercial EL systems. txt2map could outperform both these systems on both evaluation corpora. The goal of creating a geoparser that extracts more street-level references than previous systems was therefore achieved. However, the generic GCNL + Wikipedia approach seems to work almost as well as the newly presented technique.

Finally, it should be noted that both evaluation corpora are quite small compared to the standard corpora in NLP literature. By limiting the tests to a subset of the selected toponyms, the effective corpus size was decreased further. Since the tested system is rule-based and the chosen corpora represent the use case in mind quite well, the measured results still make a strong argument for the chosen approach. The achieved accuracy values, however, should not be interpreted as real-world performance indicators.

Chapter 7

Conclusion

This chapter briefly summarizes the content and learnings of this thesis. It then discusses future steps that can be taken to validate and improve the presented approach.

7.1 Summary

The thesis presented a new approach to street-level geoparsing. The process involves three steps:

1. Extract local anchor points (i.e. references to populated places)
   (a) Use general-purpose NER to recognize toponyms
   (b) Retrieve resolution candidates from GeoNames
   (c) Use heuristics to select a candidate for each toponym
   (d) Resolve toponyms to the selected GeoNames entry
   (e) Use city-level resolutions as local anchor points
2. Generate street-level gazetteers for local anchor points
   (a) Define areas around each anchor points
   (b) Load named elements within those areas from OpenStreetMap
   (c) Construct lookup gazetteers
3. Use the gazetteers to recognize and resolve local toponyms
   (a) Scan the document for names listed in the gazetteers
   (b) Resolve matched toponyms to corresponding OpenStreetMap elements

This technique makes use of several well-known methods from geoparsing literature. Gazetteers have been used for geoparsing for a long time. Using general-purpose NER for toponym recognition is a common choice in modern systems. The employed disambiguation heuristics (one-referent-per-sense, default sense, ontological similarity) have been described several times before literature. And both GeoNames and OpenStreetMap have been used as reference ontologies in previous work. The novel contribution of this thesis is the dynamic creation of local gazetteers that reflect a document's geographic scope (step 2). This step makes it possible to solve the problem of street-level geoparsing with the methods previously used for global geoparsing.

The thesis then presented txt2map, an open-source geoparsing system that implements the described technique. The system is modeled as a document processing pipeline, where each step of the pipeline adds additional annotation layers to the document. The system consists of four separate modules:

1. NER Module
2. Reclassification Module
3. Global Geoparsing Module
4. Local Geoparsing Module

Modules 1 to 3 are responsible for local anchor extraction (step 1) and module 4 is responsible for gazetteers construction and matching (steps 2 and 3). The reclassification module was added because general-purpose NER tools reliably detect entity names, but often misclassify toponyms as other entity types (e.g. organizations). It was shown that this system recognized and resolves more street-level toponyms than other state-of-the-art geoparsing and EL systems.

7.2 Learnings

This thesis examined previous and new approaches to street-level geoparsing. First is was demonstrated that previously available geoparsing and EL systems are not capable of resolving highly local toponyms like street names. This was attributed to three systematic limitations:

1. The used recognition methods miss or misclassify street-level toponyms
2. The used ontologies do not cover street-level entities
3. The used disambiguation methods lack local context

The thesis then presented a new geoparsing technique designed to overcome these limitations. Its underlying hypothesis was that a document's geographic scope can be inferred from the geopolitical entities it mentions. The txt2map system, which implements the proposed technique, was used to verify this hypothesis. Through experiments on geo-annotated corpora it could be shown that, on the street level, txt2map performs better than the chosen reference systems. These results suggest that the hypothesis is accurate.

7.3 Future Work

The system presented here is a proof-of-concept. Now that the approach has been verified, the next steps will be to improve its components and further measure and study its performance characteristics. Eventually, the ambition of this project is to offer a system that application developers can use out-of-the-box to extract street-level references from arbitrary sources.

Many aspects of the txt2map systems can be optimized. An important feature for practical use is support for non-canonical toponym forms. Approaches to handle abbreviated and misspelled toponyms could be taken from work on social media geoparsing and NER (see for example [Al-Olimat et al., 2018]). GeoNames and OpenStreetMap further provide alternative names for certain entities. By adding such variants to the local gazetteers, the overall geoparsing performance could be improved.

The combination of NER tool and reclassifier for toponym recognition can be improved as well. Gritta et al. [2018a] has shown that a custom CRF-based recognizer can outperform even the Google Cloud Natural Language API. The use of such a model would make the whole reclassification module obsolete, as its sole purpose is to post-optimize general-purpose NER tools that distinguish multiple entity classes. A less radical improvement would be to determine which types of toponyms get misclassified most frequently by general-purpose NER tools and then adjust the reclassification module to cover exactly these toponyms.

Finally, the current system only makes use of a single heuristic to resolve non-default toponym senses. Although the chosen heuristic proved to be very useful, it is clear that the combination of multiple heuristics can improve overall performance (see for example [Lieberman and Samet, 2012]). Finding the optimal weighting of different heuristics would then again be a task for machine learning.

To better understand the capabilities of the txt2map system, it would be interesting to see how the system performs on other classes of documents. The corpora used to evaluate geoparsing performance here consist exclusively of newspaper articles. While these texts are a good benchmark for geoparsing in general, tests with other document classes could reveal limitations that those tests failed to detect. For example, some applications might require the system to geoparse novels and short stories. To support this use case, it should be verified that the chosen approach also performs well for longer texts with fewer location references. This would, however, require the creation of a new corpus.

As soon as there are practical applications available, the user acceptance of the produced results should be measured as well. Empirical methods could be used to collect such metrics.

The possible applications of street-level geoparsers are diverse. They can be used to index large collections of documents, not just by city but by specific locations within cities. Such indices can be used to filter a collection by neighborhood or point of interest. Another possible application is news stream analysis. The txt2map can for example extract streets mentioned in traffic reports, restaurants recommended in food blogs, meeting points popular in certain online communities, or areas often mentioned in crime sections of newspapers. In general, street-level geoparsers allow applications to establish connections between texts and places. By following these connections, their users can discover new locations, learn new things and meet new people.

Bibliography

Adams, B., McKenzie, G., and Gahegan, M. (2015). FrankenPlace: interactive thematic mapping for ad hoc exploratory search. In *Proceedings of the 24th international conference on world wide web*, pages 12–22. International World Wide Web Conferences Steering Committee.

Al-Olimat, H. S., Thirunarayan, K., Shalin, V., and Sheth, A. (2018). Location name extraction from targeted text streams using gazetteer-based statistical language models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1986–1997, Santa Fe, New Mexico. Association for Computational Linguistics.

Alex, B. (2017). Geoparsing english-language text with the edinburgh geoparser. *Programming Historian*.

Amitay, E., Har'El, N., Sivan, R., and Soffer, A. (2004). Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280. ACM.

Apache Software Foundation (2019). Apache OpenNLP Developer Documentation (Version 1.9.1). `https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html`. Accessed: 2019-10-26.

Batista, D. S., Silva, M. J., Couto, F. M., and Behera, B. (2010). Geographic signatures for semantic retrieval. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*, page 19. ACM.

Bender, O., Och, F. J., and Ney, H. (2003). Maximum entropy models for named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 148–151. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bengio, Y., Simard, P., Frasconi, P., et al. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Berico Technologies (2016). CLAVIN: Cartographic location and vicinity indexer. `https://clavin.bericotechnologies.com`. Accessed: 2019-10-26.

Boden, M. (2002). A guide to recurrent neural networks and backpropagation. *the Dallas project, SICS Technical Report T2002:03*.

Brants, T. (2000). Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics.

Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Chinchor, N. and Robinson, P. (1997). Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29, pages 1–21.

Chiticariu, L., Li, Y., and Reiss, F. R. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA. Association for Computational Linguistics.

Chiu, J. P. and Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Cohen, W. W. (2004). Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Cunningham, H., Maynard, D., and Bontcheva, K. (2014). Developing Language Processing Components with GATE Version 8. Technical report, University of Sheffield Department of Computer Science.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: An architecture for development of robust HLT applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 168–175, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*.

DeLozier, G., Baldridge, J., and London, L. (2015). Gazetteer-independent toponym resolution using geographic word profiles. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Durrett, G. and Klein, D. (2014). A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the association for computational linguistics*, 2:477–490.

Explosion AI (2019). spacy models. `https://spacy.io/models`. Accessed: 2019-10-26.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics.

Gelernter, J. and Balaji, S. (2013). An algorithm for local geoparsing of microtext. *GeoInformatica*, 17(4):635–667.

Gelernter, J. and Zhang, W. (2013). Cross-lingual geo-parsing for non-structured data. In *Proceedings of the 7th Workshop on Geographic Information Retrieval*, pages 64–71. ACM.

Goldberg, D. W., Wilson, J. P., and Knoblock, C. A. (2007). From text to geographic coordinates: the current state of geocoding. *URISA*, 19(1):33.

Goldberg, Y. and Nivre, J. (2012). A dynamic oracle for arc-eager dependency parsing. In *COLING*.

Grice, H. P. (1975). Logic and conversation. In Cole, P. and Morgan, J., editors, *Syntax and Semantics*, volume 3, pages 41—58. Academic Press, New York, NY.

Grishman, R. and Sundheim, B. (1995). Design of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, pages 1–11. Association for Computational Linguistics.

Gritta, M., Pilehvar, M. T., and Collier, N. (2018a). A pragmatic guide to geoparsing evaluation. *arXiv preprint arXiv:1810.12368*.

Gritta, M., Pilehvar, M. T., and Collier, N. (2018b). Which melbourne? augmenting geocoding with maps. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1285–1296.

Gritta, M., Pilehvar, M. T., Limsopatham, N., and Collier, N. (2018c). What's missing in geographical parsing? *Language Resources and Evaluation*, 52(2):603–623.

Grover, C. and Tobin, R. (2006). Rule-based chunking and reusability. In *LREC*, pages 873–878.

Hahmann, S. and Burghardt, D. (2013). How much information is geospatially referenced? networks and cognition. *International Journal of Geographical Information Science*, 27(6):1171–1189.

Hoang, T. B. N., Moriceau, V., and Mothe, J. (2017). Predicting locations in tweets. In *Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing (CLICLing 2017)*.

Hobbs, J. R. (1985). On the coherence and structure of discourse. Technical report, CSLI Stanford, CA.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 91(1).

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Honnibal, M. and Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

Inkpen, D., Liu, J., Farzindar, A., Kazemi, F., and Ghazi, D. (2015). Detecting and disambiguating locations mentioned in twitter messages. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 321–332. Springer.

Ji, H. and Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 1148–1158. Association for Computational Linguistics.

Ji, Z., Sun, A., Cong, G., and Han, J. (2016). Joint recognition and linking of fine-grained locations from tweets. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1271–1281. International World Wide Web Conferences Steering Committee.

Ju, Y., Adams, B., Janowicz, K., Hu, Y., Yan, B., and McKenzie, G. (2016). Things and strings: improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling. In *European Knowledge Acquisition Workshop*, pages 353–367. Springer.

Jurafsky, D. and Martin, J. H. (expected 2019). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 3. Pearson.

Kamalloo, E. and Rafiei, D. (2018). A coherent unsupervised model for toponym resolution. In *Proceedings of the 2018 World Wide Web Conference*, pages 1287–1296. International World Wide Web Conferences Steering Committee.

Karimzadeh, M., Pezanowski, S., MacEachren, A. M., and Wallgrün, J. O. (2019). GeoTxt: A scalable geoparsing system for unstructured text geolocation. *Transactions in GIS*, 23(1):118–136.

Khashabi, D., Sammons, M., Zhou, B., Redman, T., Christodoulopoulos, C., et al. (2018). CogCompNLP: Your Swiss Army Knife for NLP. In *11th Language Resources and Evaluation Conference.*

Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th ICML*, pages 282—289. Morgan Kaufmann, San Francisco, California.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. (2015). DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

Leidner, J. L. (2007). Toponym resolution in text: Annotation, evaluation and applications of spatial grounding. *SIGIR Forum*, 41(2):124–126.

Lieberman, M. D. and Samet, H. (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*, pages 731–740. ACM.

Lieberman, M. D., Samet, H., and Sankaranarayanan, J. (2010). Geotagging with local lexicons to build indexes for textually-specified spatial data. In *2010 IEEE 26th international conference on data engineering (ICDE 2010)*, pages 201–212. IEEE.

Ling, X., Singh, S., and Weld, D. S. (2015). Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Lucy, L. and Gauthier, J. (2017). Are distributional representations ready for the real world? evaluating word vectors for grounded perceptual meaning. *arXiv preprint arXiv:1705.11168*.

Luo, G., Huang, X., Lin, C.-Y., and Nie, Z. (2015). Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

Malmasi, S. and Dras, M. (2015). Location mention detection in tweets and microblogs. In *Conference of the Pacific Association for Computational Linguistics*, pages 123–134. Springer.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

McDonald, D. (1996). Internal and external evidence in the identification and semantic categorization of proper names. In Boguraev, B. and Pustejovsky, J., editors, *Corpus Processing for Lexical Acquisition*, chapter 2, pages 21–39. The MIT Press, Cambridge, MA.

Melo, F. and Martins, B. (2015). Geocoding textual documents through the usage of hierarchical classifiers. In *Proceedings of the 9th Workshop on Geographic Information Retrieval*, page 7. ACM.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM.

Middleton, S. E., Middleton, L., and Modafferi, S. (2013). Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, 29(2):9–17.

Mihalcea, R. and Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

Mikheev, A., Moens, M., and Grover, C. (1999). Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*, pages 1–12.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. A. (2015). Computing numeric representations of words in a high-dimensional space. US Patent 9,037,464.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nivre, J. (2008). Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666.

Pasley, R. C., Clough, P. D., and Sanderson, M. (2007). Geo-tagging for imprecise regions of different sizes. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 77–82. ACM.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Piskorski, J. and Yangarber, R. (2013). Information extraction: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 23–49. Springer.

Pradhan, S., Moschitti, A., Xue, N., Ng, H. T., Björkelund, A., Uryupina, O., Zhang, Y., and Zhong, Z. (2013). Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Purves, R. S., Clough, P., Jones, C. B., Hall, M. H., Murdock, V., et al. (2018). Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends® in Information Retrieval*, 12(2-3):164–318.

Qin, T., Xiao, R., Fang, L., Xie, X., and Zhang, L. (2010). An efficient location extraction algorithm by leveraging web contextual information. In *proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 53–60. ACM.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *CoNLL*.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing*, pages 133—142.

Rauch, E., Bukatin, M., and Baker, K. (2003). A confidence-based framework for disambiguating geographic terms. In *Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references*, pages 50–54.

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Edinburgh, United Kingdom. Association for Computational Linguistics.

Samet, H. (2014). Using minimaps to enable toponym resolution with an effective 100rate of recall. In *Proceedings of the 8th Workshop on Geographic Information Retrieval*, GIR '14, pages 9:1–9:8, New York, NY, USA. ACM.

Santos, J., Anastácio, I., and Martins, B. (2015). Using machine learning methods for disambiguating place references in textual documents. *GeoJournal*, 80(3):375–392.

Sil, A. and Yates, A. (2013). Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2369–2374. ACM.

So, D. R., Liang, C., and Le, Q. V. (2019). The evolved transformer. *CoRR*, abs/1901.11117.

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

Strubell, E. and McCallum, A. (2017). Dependency parsing with dilated iterated graph CNNs. In *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pages 1–6, Copenhagen, Denmark. Association for Computational Linguistics.

Strubell, E., Verga, P., Belanger, D., and Mccallum, A. (2017). Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680.

Sultanik, E. A. and Fink, C. (2012). Rapid geotagging and disambiguation of social media text via an indexed gazetteer. *Proceedings of ISCRAM*, 12:1–10.

Sutton, C., McCallum, A., et al. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Tobin, R., Grover, C., Byrne, K., Reid, J., and Walsh, J. (2010). Evaluation of georeferencing. In *proceedings of the 6th workshop on geographic information retrieval*, page 7. ACM.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for computational Linguistics.

Tsai, C.-T. and Roth, D. (2016). Illinois cross-lingual wikifier: Grounding entities in many languages to the english wikipedia. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 146–150.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wallgrün, J. O., Karimzadeh, M., MacEachren, A. M., and Pezanowski, S. (2018). GeoCorpora: Building a corpus to test and train microblog geoparsers. *International Journal of Geographical Information Science*, 32(1):1–29.

Wang, C., Xie, X., Wang, L., Lu, Y., and Ma, W.-Y. (2005). Detecting geographic locations from web resources. In *Proceedings of the 2005 workshop on Geographic information retrieval*, pages 17–24. ACM.

Weischedel, R. and Brunstein, A. (2005). Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 112.

Weissenbacher, D., Tahsin, T., Beard, R., Figaro, M., Rivera, R., Scotch, M., and Gonzalez, G. (2015). Knowledge-driven geospatial location resolution for phylogeographic models of virus migration. *Bioinformatics*, 31(12):i348–i356.

Yampolskiy, R. V. (2013). Turing test as a defining feature of ai-completeness. In *Artificial intelligence, evolutionary computing and metaheuristics*, pages 3–17. Springer.

Yang, J., Liang, S., and Zhang, Y. (2018). Design challenges and misconceptions in neural sequence labeling. *arXiv preprint arXiv:1806.04470*.

Zhang, W. and Gelernter, J. (2014). Geocoding location expressions in twitter messages: A preference learning method. *Journal of Spatial Information Science*, 2014(9):37–70.

Zhang, Y. and Nivre, J. (2011). Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193. Association for Computational Linguistics.