Location Name Extraction from Targeted Text Streams using Gazetteer-based Statistical Language Models

Hussein S. Al-Olimat, Krishnaprasad Thirunarayan, Valerie L. Shalin and Amit Sheth
Kno.e.sis Center, Wright State University, Dayton, OH
{hussein;tkprasad;valerie;amit}@knoesis.org

Abstract

Extracting location names from informal and unstructured social media data requires the identification of referent boundaries and partitioning compound names. Variability, particularly *systematic* variability in location names (Carroll, 1983), challenges the identification task. Some of this variability can be anticipated as operations within a statistical language model, in this case drawn from gazetteers such as OpenStreetMap (OSM), Geonames, and DBpedia. This permits evaluation of an observed n-gram in Twitter targeted text as a legitimate location name variant from the same location-context. Using n-gram statistics and location-related dictionaries, our Location Name Extraction tool (LNEx) handles abbreviations and automatically filters and augments the location names in gazetteers (handling name contractions and auxiliary contents) to help detect the boundaries of multi-word location names and thereby delimit them in texts.

We evaluated our approach on 4,500 event-specific tweets from three targeted streams to compare the performance of LNEx against that of ten state-of-the-art taggers that rely on standard semantic, syntactic and/or orthographic features. LNEx improved the average F-Score by 33-179%, outperforming all taggers. Further, LNEx is capable of stream processing.[1]

1 Introduction

In context-aware computing, location is a fundamental component that supports a wide-range of applications (Hazas et al., 2004; Licht et al., 2017). During natural disasters, location is crucial for situational awareness during disaster response (Son et al., 2008). When available, targeted streams of social media data are therefore particularly valuable for disaster response (Munro, 2011)[2]. For example, the tweet "water level in Ganapathy Colony is around 2 m" refers to a location. But, unless we know where "Ganapathy Colony" is, the water level data cannot enhance situational awareness and inform disaster response applications such as storm surge modeling or forecasting.

However, pragmatic influences on writing style shorten names to reduce redundant content in social media. We call this the *location name contraction problem*. For example, "Balalok School", appears in the Chennai flood tweets in contrast to the full gazetteer name–"Balalok Matriculation Higher Secondary School". Carroll (1983) examined the complex phenomenon of alternate name forms (called Nameheads). He distinguishes between four shortening processes: (1) Appellation Formation, (2) Explicit Metonomy, (3) Category Ellipsis, and (4) Location Ellipsis.

Appellation Formation occurs when, for example, the author refers to the location name "The Erie Canal" as "The Canal". People may also refer to the only airport in the affected area as just "The Airport". Referring to "University of Michigan" as "Michigan" is an example of Explicit Metonomy. Common ground or shared understanding between the author and the recipient establishes the referent (Resnick et al., 1991). Both Appellation formation and Metonomy pose *disambiguation* problems, and require context such as the author's location to resolve.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/

[1] Data and the tool is available at https://github.com/halolimat/LNEx

[2] We define a *targeted stream* as a set of tweets that has the potential to satisfy an event-related information need (Piskorski and Ehrmann, 2013) crawled using keywords and hashtags, to contextualize the event (e.g., "#HarveyFlood").

In contrast, Category Ellipsis and Location Ellipsis pose *delimitation* problems that can be resolved with a statistical language model. Category Ellipsis occurs when the author strips words related to the location category (e.g., "City" from "Houston City" to become "Houston"). Location Ellipsis occurs when an author drops the specific location reference in the location name (e.g., when "New York Yankee Stadium" becomes "Yankee Stadium" or "Cars India - Adyar" becomes "Cars India").

This distinction between delimitation and disambigution is important in the location extraction literature. Entity delimitation is typically the first step of location extraction, to identify the boundaries of a location mention in the text. To address this problem, previous research (Liu et al., 2014; Malmasi and Dras, 2015; Hoang and Mothe, 2018) has applied both syntactic heuristics (using lexical cues, e.g., "*in* New Orleans") as well as semantic heuristics (i.e., content-based, for different types of locations such as *buildings* and *streets*). These heuristics have serious limitations, such as failing to delimit metonyms and location names that begin sentences (i.e., outside locative expressions, e.g., "*New Orleans* is flooded") and they cannot assist in hashtag segmentation (needed to extract locations from hashtags). Moreover, simply identifying a location name still leaves open the problem of linking the entity to a corresponding gazetteer record for geocoding. Simple fixed phrase matching with gazetteers entries, as in (Middleton et al., 2014; Malmasi and Dras, 2015), solves the linking problem, but remains vulnerable in two respects. With simple fixed phrase matching, the tendency for authors to shorten names while the gazetteers extend names, creates conflicting conditions causing poor recall. On the other hand, simply relaxing matching criteria exacerbates the disambiguation problem.

To address delimitation, we treat location names as a sequence of ordered words known as *collocations* (Manning and Schütze, 1999). Collocations are neither strictly compositional nor always atomic. We cannot identify them with grammatical rules, and fixed phrase matching is not reliable for longer names. Fortunately, the gazetteer provides a resource to establish region-specific naming regularities. Given a region-specific gazetteer, which retains the same location-context as the text, we can construct a statistical model of the *token sequences* it contains. However, current gazetteers are overly specific in two respects. First, consistent with (Carroll, 1983), they do not always represent Category Ellipsis and Location Ellipsis. To mimic these processes, we judiciously apply a skip-gram method to token sequences in the gazetteers, thereby including, for example, "Balalok School" as a variant of the complete name (which is a special case of Category Ellipsis where we retain only the last category token and drop the other ones). Second, we eliminate auxiliary or ambiguous gazetteer content (e.g., "George, Washington") that would otherwise threaten recall.

Here we answer the research question: ***Can we accurately and rapidly spot location mentions in text solely relying on a statistical language model synthesized from augmented and filtered region-specific gazetteers?*** Although LNEx works on a targeted Twitter stream collected using event-specific keywords, it does not rely on rarely available tweets geo-coordinates and it does not need any supervision (i.e., training data). It is well-suited for stream processing, and needs only freely available data. Our contributions include:

1. A method for preparing high-quality gazetteers from online open data, such as OSM, Geonames, and DBpedia, and deriving a language model from them; and a comprehensive analysis of the contribution of gazetteer quality to overall performance.

2. A referent corpus representing the full scope of location name extraction challenges and a challenge-based categorization of place names found in the corpora resulting from targeted streams. We annotate three different Twitter streams from flooding events in three different locations: the 2015 Chennai flood, the 2016 Louisiana flood, and the 2016 Houston flood, for our own evaluation and also for use by others.

3. A demonstration that LNEx convincingly outperforms commercial-grade NER and Twitter-specific tools with at least a 33% improvement on average F-Score. Examples reveal the true challenges of location name extraction and the locus of tool failure in the face of these challenges.

LNEx provides the foundation for localizing information, and with increased availability of open data, we expect our approach based on region-specific knowledge to be widely applicable in practice.

2 The LNEx Method

We discuss the details of LNEx in four subsections. First, we present the general idea of statistical inference via n-gram models; the core of LNEx is a statistical language model consisting of a probability distribution over sequences of words (collocations) that represent location names in preexisting, region-specific gazetteers. Then we separately discuss several modifications to both gazetteers and text samples, including gazetteer augmentation and filtering, and tweet preprocessing. Finally, we illustrate the full location analysis and matching process that reliably spots location names.

2.1 Statistical Inference via n-gram Models

LNEx constructs an n-gram model from the collocations that exist *in the gazetteer* to determine the valid location names (LNs) that might appear in tweets. Given tweet content such as "texas ave is closed", the model can then check the validity of one to n-grams. From the gazetteer, "texas" and "ave" are valid gazetteer unigrams but "is" and "closed" are not. Similarly, "texas ave" is a valid and preferred bigram (over two unigrams).

Specifically, as shown in Algorithm 1, we first tokenize all location names in the gazetteer to construct the n-gram model and then save the resulting lists of unigrams, bigrams, and trigrams (Lines 2-5). Next, for bigrams and trigrams, in Lines 6-9 we create conditional frequency distributions (CFD) to count the collocations (i.e., $c(\cdot)$ in equations 1-2). Conditional probability distributions (CPD) are then constructed from the recorded n-grams using maximum likelihood estimation (MLE). We make the assumption that only the previous two words determine the probability of the next word (Markovian assumption of order two)[3]. MLE assumes zero probability values to tokens missing from the gazetteers. *This data sparsity problem is mitigated by augmenting the gazetteers with location name variants* (see Section 2.2). In lines 11-13, we determine the validity of an n-gram (the string s) using the boolean function `VALID-N-GRAM` with the help of equations 1-4, where $c(w_x^y) \equiv c(w_x w_{x+1} \dots w_y)$, w_x^y is the collocation count (i.e., the occurrences of the consecutive words, w_x to w_y), $P(w_z|w_x^y)$ is the conditional probability of a word w_z given previous collocation w_x^y, and the chain of probabilities P_1 (for unigrams), P_2 (for bigrams), and P_3 (for tri or larger grams).

$$P(w_z|w_x^y) = \frac{c(w_x^z)}{c(w_x^y)} \quad (1)$$

$$P_1 = P(w_1^1) = \frac{c(w_1)}{\sum_{i=1}^{|\text{unigrams}|} c(w_i)} \quad (2)$$

$$P_2 = P(w_1^2) = P_1 \times p(w_2 \mid w_1^1) \quad (3)$$

$$P(w_1^n) = P_2 \times \prod_{i=3}^{n} P(w_i \mid w_{i-2}^{i-1}), n \geq 3 \quad (4)$$

Algorithm 1 Language Model Generation

```
1:  procedure COMPUTE-MODEL(Gazetteer)
2:      for ln ∈ Gazetteer do
3:          unigrams ← tokenize(ln);
4:          bigrams, trigrams ← generate from unigrams;
5:      end for
6:      for n-grams ∈ {bigrams} ∪ {trigrams} do
7:          CFD ← create CFD using n-grams;
8:          CPD ← create CPD using CFD;
9:      end for
10: end procedure

11: procedure VALID-N-GRAM(string = s): boolean
12:     w_1^n = (w_1, ..., w_n) ← tokenize(s);
        return P(w_1^n) > 0          ▷ calculated using the equations (1-4);
13: end procedure
```

2.2 Gazetteer Augmentation and Filtering

We faced two primary challenges when building our language model using raw gazetteers, which are not adequately explored in (Middleton et al., 2014; Weissenbacher et al., 2015):

1. **Conditional Collocation Contractions**: Some atomic n-gram location names (collocations) cannot be shortened, e.g., "New York". However, contraction does preserve the meaning of some longer names, especially when the first and the last words denote a specific part and a generic part respectively, such as in "Balalok School".

2. **Auxiliary and Spurious Content**: Gazetteer entries may contain extraneous content that can cause location matching to fail. Cleaning such entries can improve matching reliability (see Table 1).

[3] We found in our dataset that roughly 98% of location mentions in tweets have less than three words.

To address these challenges, we took inspiration from Category Ellipsis (for collocation contraction) and Location Ellipsis (for filtering the auxiliary content) as follows:

1. **Skip-grams**: Given a location name $t_1 \dots t_n$, we retain t_1 and t_n while varying $t_2 \dots t_{n-1}$. To avoid adding "City York" as a legitimate variant of the location name "City College of New York", we require t_n to be a location category name (e.g., building, road). Therefore, "Balalok Matriculation Higher Secondary School" generates {`Balalok School`, `Balalok Secondary School`, ...}. This technique results in a small number of contractions that are either useful collocations or are too random to cause many false positives (Guthrie et al., 2006).

2. **Filtering**: To address bracketed auxiliary content, we compiled a generic list of phrases to remove specific words on a case-by-case basis (e.g., 1-2 in Table 1). The remaining bracketed names are deemed legitimate alternatives (e.g., 3-4 in Table 1). We treat hyphenated location names as Location Ellipsis and split them on the hyphen and add the two splits (e.g., 5 in Table 1). We expect that the majority of these location names represent a partonomy relationship where the hyphen may be read as a "part of" relation between split tokens. We do not add the second token as a variant when it already exists in the gazetteer on its own as location name entity (e.g., "Hammond" in "Pilot - Hammond").

	Content Description	Example Gazetteer Record
1	Descriptive Tags	(Private Road)
2	Life-cycle/Status Tags	Little Rock School (historical)
3	Alternative/Old Names	Scenic Road (Frontage Road)
4	Acronyms	International House of Pancakes (IHOP)
5	Hyphenations	Cars India - Adyar, Pilot - Hammond

Table 1: Extraneous text in raw gazetteers

These two methods augment and filter partial OSM, Geonames, and DBpedia gazetteers sliced from the original sources using a bounding box. Further, we can attach the metadata of the original location name to the generated variants. Moreover, by treating derived names as synonyms for existing names, we avoid creating additional demands on disambiguation or equivalencing. We add the derived, variant location name to the gazetteer as long as it does not collide with an existing location name. Additional filtering of proposed variants is required to prevent false alarms. Similar to the use case in (Weissenbacher et al., 2015; Gelernter and Balaji, 2013), we compiled a list of 11,203 words including 678 inseparable bigrams, such as "Building A", as gazetteer stop words. This list also includes unusual location names (e.g., "Boring" in Maryland and "Why" in Arizona) and proper nouns (e.g., "James" in Mississippi) that could appear as non-location tokens. We then eliminate from all gazetteers the location names that overlap with our gazetteer stop words to reduce false positives.

2.3 Tweet Preprocessing

To complement the gazetteer preprocessing, we also require potentially non-trivial tweet preprocessing. We start by removing the retweet handles, URLs, non-ASCII characters, and all user mentions. Then, we tokenize tweets using TweetMotif's Twokenizer (O'Connor et al., 2010), which treats hashtags, mentions, and emoticons as a single token. We do not tokenize on periods (e.g., "U.S.").

Hashtag Segmentation: In our datasets, on average, around 29% of the hashtags include location names. Excluding hashtags used to crawl the data, around 17% of the unique hashtags contain locations. As the number of locations in hashtags is significant, similar to (Malmasi and Dras, 2015), we adopted a statistical word segmentation algorithm to break hashtags for location spotting (Norvig, 2009).

Spelling Correction: We consider a tweet token as misspelled if it is an out-of-vocabulary token, where the vocabulary is gazetteer words and a large English vocabulary word list[4]. LNEx corrects all misspelled tokens using the Symmetric Delete Spelling Correction algorithm (SymSpell)[5] that is six orders of magnitude faster than Norvig's spelling corrector (Norvig, 2009), which was used by (Gelernter and Zhang, 2013) in their location extraction tool. As we shall see, spelling correction has only a small influence on system accuracy.

[4] `https://github.com/norrissoftware/words3`
[5] `https://github.com/wolfgarbe/symspell`

2.4 Extracting Location Names using LNEx

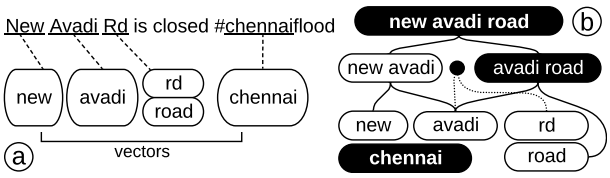

Figure 1: Extracting Locations using LNEx

After the modifications to gazetteers and texts, LNEx extracts locations as illustrated in Fig. 1. In ⓐ, LNEx reads the raw tweet text, preprocesses it (as in Section 2.3) starting with case-folding. After tokenizing the tweet, the hashtag segmenter breaks hashtags into tokens. Later, stop words are used to split a tweet into consecutive word fragments where each tweet split of size n can have zero to n potential location names. We custom build the tweet stop list starting with around 890 words[6] excluding the gazetteer unigrams.

LNEx now takes each tweet split and converts its tokens into a vector of tokens v using two dictionaries: the USPS street suffixes dictionary[7] and the English OSM abbreviations dictionary[8]. This adds possible expansions and abbreviations of a token (e.g., "Rd" to "Road", and vice versa). This overcomes the lexical variations between location mentions in tweets and their corresponding gazetteer entries.

In ⓑ, the language model is used to find the valid n-grams from the Cartesian product of the consecutive vectors. It builds a bottom up tree for each tweet split starting from 1 to n-grams by gluing the consecutive tokens together if they represent a valid segment in the gazetteer. We improve the speed of the algorithm significantly by splitting the tweet and eliminating invalid n-grams (i.e., n-grams with zero probability values). LNEx then selects a subset of valid n-grams from the tree; for the overlapping n-grams, we prefer the longest full mentions (e.g, "New Avadi Road" over "Avadi Road") and keep both if they are of the same length. When full location names appear inside partial ones, we keep only full names (e.g., extracting "Louisiana" from "The Louisiana").

Time and Space Complexities: LNEx extracts and links a full location mention to its corresponding gazetteer entry through a simple dictionary lookup that takes constant time $\mathcal{O}(1)$. The location extraction time is bounded by the time for creating the bottom up tree of tokens which takes $\mathcal{O}(|v|^s)$ where $|v|$ is the length of the longest vector of token synonyms (i.e., all the expansions and abbreviations of a token) and s is the largest number of tokens with synonyms in a location name. Luckily, splitting the tweet into smaller fragments using stop words significantly lowers the asymptotic growth of the algorithm, thereby enabling stream processing. In practice, for our dictionaries and gazetteers, $|v| \leq 4$ and $s \leq 3$. So, a pessimistic upper bound on the number of candidates for each location (though rarely realized) is 4^3. The space complexity of the method is bounded by the product of number of gazetteer entries, L, and the number of variants of a location name (Skip-gram method 1), that is, 2^{m-2}, where m is the number of tokens in a location name. Effectively, the space complexity is $\mathcal{O}(L.2^{m-2})$ where typically, $2 \leq m \leq 5$. Further, according to our tests, LNEx needed only up to 650 MB of memory to initialize and is able to process, on average, 200 tweets per second.

3 Experimental Results

To demonstrate the effectiveness of our context-aware location extractor, we used a set of event-specific hashtags and keywords to collect 4,500 geographically limited, disaster-related tweets from three different targeted streams corresponding to floods in Chennai, Louisiana, and Houston. Below, we categorize and annotate these tweets used for benchmarking each component of LNEx, and for comparing LNEx with other state-of-the-art tools for the location extraction task.

3.1 Benchmarking and Annotations

Consistent with the problem of determining whether a location mention is inside the area of interest, our benchmark categorization scheme is not content based as in (Matsuda et al., 2015; Gelernter and Balaji, 2013). To better identify and characterize the challenges in extracting location names accurately, our

[6] http://www.ranks.nl/stopwords
[7] http://pe.usps.gov/text/pub28/28apc_002.htm
[8] wiki.openstreetmap.org/wiki/Name_finder:Abbreviations

annotation scheme is based on where these locations lie in relation to the area of interest. For example, with respect to Chicago, IL, USA:

1. in**LOC**: Locations inside the area of interest, (e.g., Millennium Park or Burlington Ave.)
2. out**LOC**: Locations outside the area of interest, (e.g., Central Park, 5th Ave, New York.)
3. amb**LOC**: Ambiguous locations that need context for identification, (e.g., "our house")

In contrast to (Matsuda et al., 2015; Gelernter and Balaji, 2013), our categorization is not based on location types (e.g., buildings, facilities, schools) but on the relative position (i.e., inLOC or outLOC) and the nature of the location mention (i.e., inLOC or ambLOC). This approach identifies the true scope of challenges in extracting location names. Other schemes that annotate for a limited set of location types, such as "Geoparse Twitter Benchmark Dataset" (Middleton et al., 2014), miss obvious location mentions in tweets, such as "New Zealand" and "Christchurch", making the dataset incompatible for testing the tools mentioned in this paper including LNEx.[9]

Tweet Annotations Figure 2 shows an example of manual annotation from the Louisiana flood tweets using the BRAT tool (Stenetorp et al., 2012). It allows us to define search functionalities and additional resources for the annotators to use such as Google Maps.

Figure 2: Example Annotations using BRAT

The annotators annotated three datasets: the 2015 Chennai flood, the 2016 Louisiana flood, and the 2016 Houston flood. In Chennai, they spotted 4,589 location names (75% inLOC, 4% outLOC, and 21% ambLOC); in Louisiana, 2,918 (66% inLOC, 13% outLOC, and 22% ambLOC); and in Houston, 4,177 (66% inLOC, 7% outLOC, and 27% ambLOC). We randomly selected 1,000 tweets (500 each from Chennai and Louisiana) as a development set and the remaining 3,500 as the test set for evaluation.

3.2 Evaluation Strategy

Because BRAT records the start and the end character offsets of the annotated LNs, we evaluate the extraction task by checking the character offsets of the spotted location name in comparison with the annotated data. We used the standard comparison metrics: Precision, Recall, and the balanced F-Score. In the case of overlapping or partial matches, we penalize all tools by adding $\frac{1}{2}FP$ (False Positive) and $\frac{1}{2}FN$ (False Negative) to the precision and recall equations (e.g., if the tool spots "The Louisiana" instead of "Louisiana").

We evaluate all tools based on the category of the extracted location in our annotation scheme. For the inLOC mentions, we count all hits and misses of a tool and ignore all hits when the category of the extracted location is outLOC or ambLOC. However, we take a particularly conservative approach and additionally penalize LNEx for extracting location names of outLOC and ambLOC categories, counting them as false positives (FPs) as our tool is not supposed to extract these.

Spell Checking: This led to 1% increase in recall but the F-Score decreased by 2% on average due to the influence of increased false positives on precision. In the final system, we excluded the spelling corrector component.

Hashtag Breaking: We evaluated the performance of the hashtag breaking component only on the hashtags that contain locations. The accuracies were 97%, 87%, and 93% for Chennai, Louisiana, and Houston respectively, reduced due to examples such as "#lawx", which was broken into "law" and "x".

Picking a Gazetteer: The augmentation and filtering of gazetteers improved the F-Scores (see Figure 3-a). After this process, combinations of gazetteer sources had similar performance where the difference between the worst and the best was around 0.02 F-Score units (see Figure 3-b). In the final system, we relied on OSM, which performed the best. Moreover, DBpedia is not focused on geographical information; therefore, it does not contain the metadata useful for the system's future use (e.g., extents and full addresses). Also, OSM has more fine-grained locations and more accurate geo-coordinates than Geonames (Gelernter et al., 2013).

[9]Our dataset can be used to test any location extraction tool by ignoring the optional additional expressivity, which makes our dataset more compatible than other available ones.

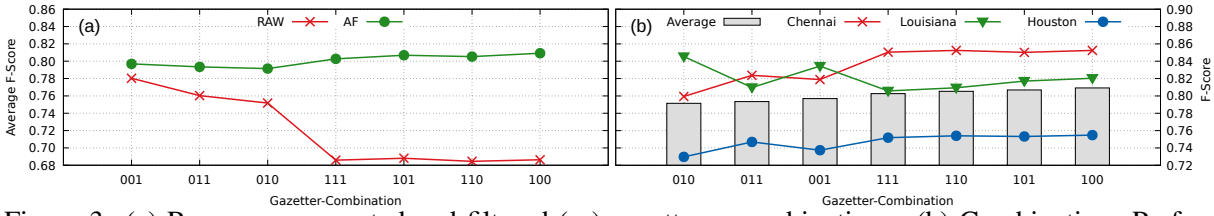

Figure 3: (a) Raw vs. augmented and filtered (AF) gazetteers combinations. (b) Combinations Performance. Each of the seven combinations is a subset of {OSM Geonames DBpedia}.

Google NLP	Location, Organization
OpenCalais	City, Company, Continent, Country, Facility, Organization, ProvinceOrState, Region, TVStation
DBpedia Spotlight	Place, Organization
OSU TwitterNLP	Geo-Location, Company, Facility
TwitIE-Gate	Location, Organization

Table 2: Types considered as Locations per tool

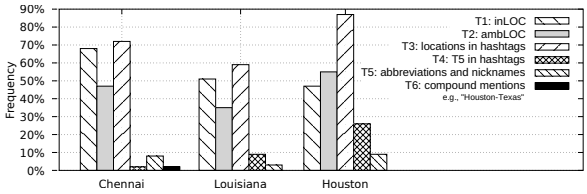

Figure 4: Random Sample Evaluation.

Comparing LNEx with other tools: We compared LNEx with the following tools:

1. **Commercial Grade**: Google NL[10], OpenCalais[11], and Yahoo! BOSS PlaceFinder[12]. All of these tools have REST APIs and are black box tools that use Machine Learning.
2. **General Purpose NER**: Stanford NER and OpenNLP Name Finder. Stanford NER learns a linear chain Conditional Random Field (CRF) sequence model (Finkel et al., 2005), while OpenNLP uses the maximum entropy (ME) framework (Bender et al., 2003). We trained both tools interchangeably on our annotated datasets in addition to all the data from W-NUT '16[13] while retaining the annotations and unifying the classes we consider as locations (i.e., geo-loc, company, facility) into one type. We used LNEx-OSM gazetteer's features while training Stanford NER. Additionally, we used DBpedia Spotlight (Mendes et al., 2011).
3. **Twitter NLP**: OSU Twitter NLP (Ritter et al., 2011) and TwitIE-Gate (Bontcheva et al., 2013). Both are pipelined systems of POS-tagging followed by NER. TwitIE-GATE also supports normalization, gazetteer lookup, and regular expression-based tagging. For fair comparison, we also augmented them with LNEx OSM gazetteers.
4. **Twitter Location Extraction**: Geolocator 3.0 (Gelernter and Zhang, 2013) and Geoparsepy (Middleton et al., 2014). Geolocator 3.0 uses a tweet-trained CRF classifier and other rule-based models to extract street names, building names, business names, and unnamed locations (i.e., location names containing a category such as "School").

All tools have been evaluated using the same metrics and on the same annotated data. In the case of hashtags, we count all hits for all tools and when a tool missed, we penalized only the ones that were designed to break hashtags (namely, TwitIE-Gate and LNEx). Additionally, we consider all spotted mentions from PlaceFinder, Geolocator 3.0, Geoparsepy, Stanford NER, and OpenNLP as location names but, for other tools, we consider only the entity types in Table 2 as location entities.

Fig. 4 shows that the prevalence of various challenges differ in the three corpora. Nevertheless, LNEx outperformed all other tools on all

	Chennai			Louisiana			Houston			AVG
	P	R	F	P	R	F	P	R	F	F
Google NLP	0.40	0.49	0.44	0.55	0.75	0.64	0.39	0.51	0.44	0.51
OpenCalais	0.43	0.10	0.17	0.81	0.77	0.78	0.62	0.35	0.45	0.47
DBpedia Spotlight	0.31	0.44	0.36	0.57	**0.88**	0.70	0.35	0.53	0.42	0.50
Yahoo! PLaceFinder	0.67	0.39	0.49	0.83	0.80	0.81	0.64	0.42	0.50	0.61
Stanford NER	0.72	0.29	0.41	0.78	0.42	0.55	0.74	0.32	0.45	0.47
OpenNLP	0.55	0.15	0.24	0.62	0.19	0.29	0.60	0.23	0.34	0.29
OSU TwitterNLP	0.74	0.40	0.52	**0.84**	0.69	0.76	0.66	0.39	0.49	0.59
TwitIE-Gate	0.51	0.36	0.43	0.66	0.84	0.74	0.35	0.39	0.37	0.52
Geolocator 3.0	0.43	0.54	0.48	0.32	0.71	0.44	0.38	0.58	0.46	0.46
Geoparsepy	0.41	0.28	0.33	0.45	0.72	0.55	0.44	0.46	0.45	0.45
LNEx-RawGaz	0.80	0.78	0.79	0.51	0.80	0.62	0.63	0.66	0.64	0.69
LNEx-AFGaz	**0.91**	**0.80**	**0.85**	0.83	0.81	**0.82**	**0.87**	**0.67**	**0.76**	**0.81**

Table 3: Tools vs. LNEx with a raw (RawGaz) or augmented and filtered gazetteer (AFGaz).

[10] https://cloud.google.com/natural-language/
[11] http://www.opencalais.com/
[12] https://developer.yahoo.com/boss/geo/
[13] http://noisy-text.github.io/2016

datasets in terms of F-Score, and the average F-Score (see Table 3). LNEx showed stability on the test and development sets from Louisiana with only a 0.2% F-Score reduction and around a 2.6% reduction on the test set from Chennai.

The augmentation and filtering method significantly improved the average F-Score from 0.69 to 0.81 (See Table 3). However, limitations of the gazetteer augmentation and filtering methods did contribute to lowering precision. For example, on average, around 5% of the extracted location names were outLOC and ambLOC, mistakenly extracted from Chennai, Louisiana, and Houston tweets. Example errors include the augmentation of location names such as "The x Apartments" to "The Apartments", causing LNEx to extract the phrase "The Apartments" as an actual full location name. Fixing such limitations should contribute to around 2% F-Score improvement on average.

We trained Stanford NER and OpenNLP to emulate their use in other studies mentioned in Section 4. Their performances were calculated by interchangeably training them using three datasets at a time and testing on the fourth one (the gazetteer of the area of the test data was also used in training the Stanford NER models). We always used the W-NUT '16 dataset to train the models with more than 10,000 tweets each time.

We observed that the ill-formatted text of tweets with ungrammatical text and missing orthographic features impact the F-Score of tools we compared with LNEx. While the performance of each tool differs, we observed that Google heavily relies on orthographic features and expects grammatical texts (even though it scored a 0.38 average F-Score). Additionally, TwitIE-GATE was not always successful in extracting location names from hashtags or text even if these location names were part of the tool's gazetteers. Finally, OpenCalais extracts only well-known location names of coarser granularity than street and building levels unless a location has an attached location category (e.g., school or street).

Original Text	sou th kr koil street near Oxford school.west mambalam..
	We r lucky where I am in New Iberia. #PrayForLouisiana #lawx
	Didn't Houston have a bad flood last year now again poor htown
Manual Annotations & Types	misspelling; T6; (**sou th** **kr** **koil street**) near (**Oxford school**).(**west mambalam**)..
	T1; T3; T4; We r lucky where I am in (**New Iberia**). #PrayFor(**Louisiana**) #(**la**)wx; T5
	Didn't **(Houston)** have a bad flood last year now again poor **(htown)**; T1; T5
Google NLP	sou th kr (**koil street**) near (**Oxford**) school.west (**mambalam**)..
	We r lucky where I am in (**New Iberia**). #PrayForLouisiana #lawx
	Didn't (**Houston**) have a bad flood last year now again poor htown
OpenCalais	sou th kr koil street near (**Oxford school**).west mambalam..
	We r lucky where I am in New Iberia. #PrayForLouisiana #lawx
	Didn't (**Houston**) have a bad flood last year now again poor htown
DBpedia Spotlight	sou th kr koil street near (**Oxford**) school.west (**mambalam**)..
	We r lucky where I am in (**New Iberia**). #PrayForLouisiana #lawx
	Didn't (**Houston**) have a bad flood last year now again poor htown
Yahoo! PlaceFinder	sou (**th**) kr koil street near (**Oxford**) school.west mambalam..
	We r lucky where I am in (**New Iberia**). #PrayForLouisiana #lawx
	Didn't Houston have a bad flood last year now again poor htown
Stanford NER	sou th kr koil street near Oxford school.west mambalam..
	We r lucky where I am in (**New Iberia**). #PrayForLouisiana #lawx
	Didn't Houston have a bad flood last year now again poor htown
OpenNLP	sou th kr (**koil street**) near (**Oxford**) school.west mambalam..
	We r lucky where I am in (**New Iberia.**) #PrayForLouisiana #lawx
	Didn't Houston have a bad flood last year now again poor htown
OSU TwitterNLP	sou th kr koil street near (**Oxford**) school.west mambalam..
	We r lucky where I am in (**New Iberia**). #PrayForLouisiana #lawx
	Didn't Houston have a bad flood last year now again poor htown
TwitIE-Gate	sou th kr koil (**street**) near (**Oxford**) school.(**west mambalam**)..
	We r lucky where I am in New Iberia. #PrayForLouisiana #lawx
	Didn't Houston have a bad flood last year now again poor htown
Geolocator 3.0	(**sou th**) (**kr**) (**koil**) street near (**Oxford school**).(**west mambalam**)..
	We r lucky where I am in (**New Iberia**). #PrayForLouisiana #lawx
	Didn't (**Houston**) have a bad flood last year now again poor (**htown**)
Geoparsepy	sou (**th**) (**kr**) koil street near (**Oxford**) school.west mambalam..
	We r lucky where I am in (**New** (**Iberia**)). #PrayForLouisiana #lawx
	Didn't (**Houston**) have a bad flood last year now again poor htown
LNEx	sou th kr koil street near (**Oxford school**).(**west mambalam**)..
	We r lucky where I am in (**New Iberia**). #PrayFor(**Louisiana**) #lawx
	Didn't (**Houston**) have a bad flood last year now again poor htown

Table 4: Example tool outputs: bracketed bold text are the identified LNs and braces highlights the types from Fig. 4.

Illustrative Examples: Table 4 shows the comparative handling of three tweets one each from Chennai, Louisiana, and Houston datasets, covering most challenging cases for all the tools. The location name "Oxford school" allowed us to examine if a tool relies on capitalization for delimitation. Only OpenCalais, Geolocator and LNEx were able to extract the name correctly while the rest either partially extracted it or missed it. For example, PlaceFinder extracted "Oxford" and geocoded it with the geocodes of Oxford city in England. Although Stanford NER and OpenNLP were trained on the same datasets, OpenNLP extracted Oxford while Stanford NER did not, which suggests that the cue word "near" was insufficient evidence for Stanford NER to spot at least Oxford. Correspondingly, since "New Iberia" is a correctly capitalized full location name, almost all tools were able to

extract it. However, TwitIE-Gate missed it although it is part of the gazetteer we added to the tool, and Geoparsepy extracted Iberia in addition to the full mention, not favoring the longest mention as LNEx. OpenCalais is a black box so we don't know why it failed.

Regarding T3-T5 annotations, LNEx and TwitIE-Gate are designed to break hashtags but TwitIE-Gate was not able to extract any locations from the hashtags in the table. LNEx extracted "Louisiana" but was not able to extract "la" from "#lawx" due to the statistical method which broke the hashtag into "law" and "x" since this combination is more probable. Only Geolocator was able to extract the Houston nickname "htown". In the future, a dictionary of region-specific acronyms, abbreviations, and nicknames can augment LNEx's region-specific gazetteers.

Google NLP does not handle T6. Adding space between the dot and "west" to create "... school. west ...", results in the extraction of "west mambalam" but omits "Oxford school". Google NLP relies on capitalization, so changing the case of "s" to create "Oxford School" does help. OpenCalais cannot extract "west mambalam" despite fixing all grammatical mistakes, normalizing the orthographic features, and even introducing cue words. The tool only extracts well-known location names of coarser granularity than street and building levels unless they have an attached location category (e.g., school or street). PlaceFinder, on the other hand, tries to find geocodable location names in text. Therefore, the tool extracts "th" as the country code of Thailand and "Oxford" as the city in England. Hence, geocoding is influencing some of the mistakes of the tool.

4 Related Work

Twitter messages (tweets) lack features exploited by main stream NLP tools. Informality, ill-formed words, irregular syntax and non-standard orthographic features of tweets challenge such tools (Kaufmann and Kalita, 2010). We agree with (Baldwin et al., 2013) that some issues might be exaggerated. Indeed we found that spelling corrections only contributed to 1% recall improvement. Nevertheless, text normalization alone is insufficient for NER (Derczynski et al., 2015). Specially designed tools such as (Ritter et al., 2011; Gelernter and Zhang, 2013) use pipelined systems of POS tagging followed by NER. The latter also perform Regex tagging, normalization, and gazetteer lookup.

Relying on the orthographic features for POS tagging or Regex tagging, previous methods extract locations from the text chunks and phrases of sentences using the following techniques:

1. **Gazetteer search or n-gram matching**: Li et al. (2014) and Gelernter and Zhang (2013) use a gazetteer matching technique that relies on a segment-based inverted index. Sultanik and Fink (2012) use an exhaustive n-gram technique. Middleton et al. (2014) use location-specific gazetteers for matching phrases from tweets. TwitIE-GATE uses a gazetteer lookup component. All of these techniques do not deal with the important issue of the gazetteers' auxiliary content and noise.
2. **Handcrafted rules**: Weissenbacher et al. (2015) and Malmasi and Dras (2015) use pattern and Regex matching which rely on cue words or orthographic features for POS-tagging. TwitIE-GATE adapts rules from ANNIE (Cunningham et al., 2002) for extraction.
3. **Supervised Methods**: *Tweet-trained models*: The majority of the methods trained Stanford NER on tweets (Gelernter and Zhang, 2013; Yin et al., 2014) or retrained OpenNLP (Lingad et al., 2013). *News-trained models*: Malmasi and Dras (2015) use tools like Stanford NER and OpenNLP.
4. **Semi-supervised methods**: Ji et al. (2016) use beam search and structured perceptron for extraction and linking to Foursquare entities. However, they did not address the noise that is prevalent in such sources (e.g., "my sofa" or "our house") (Dalvi et al., 2014).

The closest works to ours are TwiNER (Li et al., 2012) and LEX (Downey et al., 2007). Both use Microsoft Web n-grams (which capture language statistics) for chunking but the former uses DBpedia for entity linking. However, our method exploits a region-specific gazetteer for delimitation and linking. Moreover, LEX worked with web data and relies heavily on capitalization.

Finally, few other methods extract locations from hashtags. Malmasi and Dras (2015) uses a statistical hashtag breaker technique similar to ours. Ji et al. (2016) removes only the # symbol and treats the hashtag as a unigram. Yin et al. (2014) uses a greedy maximal matching method for breaking. TwitIE-GATE

uses two methods for hashtag breaking: a dynamic programming-based method for finding subsequences and a camel-case-based method for tokenization.

5 Conclusions and Future Work

LNEx accurately spots locations in text relying solely on statistical language models synthesized from augmented and filtered region-specific gazetteers. It outperforms state-of-the-art techniques and mainstream location name extractors. By exploiting the knowledge in the gazetteer, we extract (delimit) location mentions and then materialize them as location metadata for future/further processing. LNEx does not employ any training and does not depend on syntactic analysis or orthographic conventions. We compensate for limitations in fixed phrase matching with gazetteer augmentation and filtering. Although we do not solve the disambiguation problem here, still the geo/geo ambiguity is reduced by preserving the spatial context through location-specific gazetteers. Furthermore, systematic gazetteer augmentation ties legitimate variants to known locations, minimizing potential ambiguity.

Certainly, LNEx does not solve all location extraction problems. It actually presents an effective precision-recall trade-off apparent in the F-Score. But as the method is driven by the linking procedure, it does not extract location names missing from gazetteers (e.g., "our house"). However, in the future, we might consider extracting them at a reduced cost by integrating LNEx with an incremental learning method (Al-Olimat et al., 2018). Additionally, a more sophisticated name model that ignores the generic parts and retains the specific parts when augmenting a location name (e.g., adding "Sam's" as a variant of "Sam's Club") can be used (Dalvi et al., 2014).

Acknowledgments

This research was partially supported by the NSF award EAR-1520870 "Hazards SEES: Social and Physical Sensing Enabled Decision Support for Disaster Management and Response". We would like to thank Jibril Ikharo for introducing us to the Nameheads work and our other colleagues from Kno.e.sis for helping us in data annotation.

References

Hussein S. Al-Olimat, Steven Gustafson, Jason Mackay, Krishnaprasad Thirunarayan, and Amit Sheth. 2018. A practical incremental learning framework for sparse entity extraction. In *The 27th International Conference on Computational Linguistics (COLING 2018)*.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *International Joint Conference on Natural Language Processing*, pages 356–364.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 148–151, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A Greenwood, Diana Maynard, and Niraj Aswani. 2013. Twitie: An open-source information extraction pipeline for microblog text. In *RANLP*, pages 83–90.

John M Carroll. 1983. Nameheads. *Cognitive science*, 7(2):121–153.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics.

Nilesh Dalvi, Marian Olteanu, Manish Raghavan, and Philip Bohannon. 2014. Deduplicating a places database. In *Proceedings of the 23rd international conference on World wide web*, pages 409–418. Association for Computing Machinery (ACM).

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49.

Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating complex named entities in web text. In *IJCAI*, volume 7, pages 2733–2739.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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

Judith Gelernter, Gautam Ganesh, Hamsini Krishnakumar, and Wei Zhang. 2013. Automatic gazetteer enrichment with user-geocoded data. In *Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, pages 87–94. Association for Computing Machinery (ACM).

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling. In *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, pages 1–4.

Mike Hazas, James Scott, and John Krumm. 2004. Location-aware computing comes of age. *Computer*, 37(2):95–97.

Thi Bich Ngoc Hoang and Josiane Mothe. 2018. Location extraction from tweets. *Information Processing & Management*, 54(2):129–144.

Zongcheng Ji, Aixin Sun, Gao Cong, and Jialong Han. 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.

Max Kaufmann and Jugal Kalita. 2010. Syntactic normalization of twitter messages. In *International conference on natural language processing, Kharagpur, India*.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730. Association for Computing Machinery (ACM).

Guoliang Li, Jun Hu, Jianhua Feng, and Kian-lee Tan. 2014. Effective location identification from microblogs. In *Data Engineering (ICDE), 2014 The Institute of Electrical and Electronics Engineers (IEEE) 30th International Conference on*, pages 880–891. The Institute of Electrical and Electronics Engineers (IEEE).

Yehoshua Zvi Licht, David Allen Turner, and Joseph Arnold White. 2017. Location context aware computing, November 30. US Patent App. 15/166,740.

John Lingad, Sarvnaz Karimi, and Jie Yin. 2013. Location extraction from disaster-related microblogs. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1017–1020. Association for Computing Machinery (ACM).

Fei Liu, Maria Vasardani, and Timothy Baldwin. 2014. Automatic identification of locative expressions from social media text: A comparative analysis. In *Proceedings of the 4th International Workshop on Location and the Web*, pages 9–16. Association for Computing Machinery (ACM).

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

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Koji Matsuda, Akira Sasaki, Naoaki Okazaki, and Kentaro Inui. 2015. Annotating geographical entities on microblog text. In *The 9th Linguistic Annotation Workshop held in conjuncion with North American Chapter of the Association for Computational Linguistics (NAACL) 2015*, page 85.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM.

Stuart E Middleton, Lee Middleton, and Stefano Modafferi. 2014. Real-time crisis mapping of natural disasters using social media. *The Institute of Electrical and Electronics Engineers (IEEE) Intelligent Systems*, 29(2):9–17.

Robert Munro. 2011. Subword and spatiotemporal models for identifying actionable information in haitian kreyol. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 68–77. Association for Computational Linguistics (ACL).

Peter Norvig. 2009. Natural language corpus data. In T. Segaran and J. Hammerbacher, editors, *Beautiful Data*, chapter 14, pages 219–242. O'Reilly Media.

Brendan O'Connor, Michel Krieger, and David Ahn. 2010. Tweetmotif: Exploratory search and topic summarization for twitter. In *ICWSM*, pages 384–385.

Jakub Piskorski and Maud Ehrmann. 2013. On named entity recognition in targeted twitter streams in polish. In *The 4th Biennial International Workshop on Balto-Slavic Natural Language Processing: ACL*, pages 84–93. Citeseer.

L.B. Resnick, J.M. Levine, and S.D. Teasley. 1991. *Perspectives on Socially Shared Cognition*. American Psychological Association.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics (ACL).

Jeongwook Son, Zeeshan Aziz, and Feniosky Pena-Mora. 2008. Supporting disaster response and recovery through improved situation awareness. *Structural survey*, 26(5):411–425.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France, April. Association for Computational Linguistics (ACL).

Evan A Sultanik and Clayton Fink. 2012. Rapid geotagging and disambiguation of social media text via an indexed gazetteer. *Proceedings of International conference on Information Systems for Crisis Response and Management (ISCRAM)*, 12:1–10.

Davy Weissenbacher, Tasnia Tahsin, Rachel Beard, Mari Figaro, Robert Rivera, Matthew Scotch, and Graciela Gonzalez. 2015. Knowledge-driven geospatial location resolution for phylogeographic models of virus migration. *Bioinformatics*, 31(12):i348–i356.

Jie Yin, Sarvnaz Karimi, and John Lingad. 2014. Pinpointing locational focus in microblogs. In *Proceedings of the 2014 Australasian Document Computing Symposium*, page 66. Association for Computing Machinery (ACM).