Automatic gazetteer enrichment with user-geocoded data

Judith Gelernter
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15323 U.S.A.
gelern@cs.cmu.edu

Gautam Ganesh
Engineering & Computer Science
University of Texas at Dallas
Richardson, TX 75080 U.S.A.
gautam199@gmail.com

Hamsini Krishnakumar
College of Engineering, Guindy
Anna University
Chennai 600025 India
hamsinikk@gmail.com

Wei Zhang
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15232 U.S.A.
weizhan1@cs.cmu.edu

ABSTRACT

Geographical knowledge resources or gazetteers that are enriched with local information have the potential to add geographic precision to information retrieval. We have identified sources of novel local gazetteer entries in crowd-sourced OpenStreetMap and Wikimapia geotags that include geo-coordinates. We created a fuzzy match algorithm using machine learning (SVM) that checks both for approximate spelling and approximate geocoding in order to find duplicates between the crowd-sourced tags and the gazetteer in effort to absorb those tags that are novel. For each crowd-sourced tag, our algorithm generates candidate matches from the gazetteer and then ranks those candidates based on word form or geographical relations between each tag and gazetteer candidate. We compared a baseline of edit distance for candidate ranking to an SVM-trained candidate ranking model on a city level location tag match task. Experiment results show that the SVM greatly outperforms the baseline.

Categories and Subject Descriptors

D.2.12 Interoperability – *Data mapping*

General Terms

Algorithms

Keywords

Geographic information retrieval (GIR), gazetteer enrichment, gazetteer expansion, approximate string match, fuzzy match, location, geo-tag

1. INTRODUCTION

Data mining of local place names may be aided by a geo-knowledge resource or gazetteer that includes neighborhoods and landmarks. GeoNames is our core gazetteer for being one of the most complete available, but it is not rich in local entries. This research contributes to enriching a gazetteer. Our research considers how we can automatically create a geo-resource that is more complete on the local level.

Our research questions

a. How do different sources of local geographic information compare? We considered OpenStreetMap and Wikimapia.
b. Can we compare mined local geographic information to the gazetteer such as to allow approximate matching in name spelling and in spatial coordinates?

Others' approach: sources of local entries

Fundamentally, gazetteer entries contain place name and geographic footprint.[1] A number of sources have been used for gazetteer entries, including from paper maps, systematic on-site collection, data mining of government, postal or tourist websites, or volunteered geographic information sites.

Authoritative toponyms and corresponding coordinates are atlas indexes or national gazetteers, or paper maps that had been digitized. But scanning from digitized map graphics is complicated by image compression and in many maps, text labels overlap with each other or with other map features and so are unreadable [6]. Nonetheless, substantial work has been done in what is called georeferencing legacy content from maps, to create crosswalks between cartographic and digital media. The National Geographic Maps' database is one such effort [4].

Toponyms have been collected manually by organizations such as the United Nations Groups of Experts on Geographic Names, and systematically at local scale by Courage Services, a U.S. company that provides cultural and geographic research.[2]

GEOCROWD '13:, November 05-08 2013, Orlando, FL, USA Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-2528-8/13/11&#0133;$15.00.
http://dx.doi.org/10.1145/2534732.2534736

[1] Additional attributes per entry might be alternate or colloquial names in the original language (endonyms), and the same name in other languages (exonyms), place population, and spatial hierarchy (city, country).

[2] http://unstats.un.org/unsd/geoinfo/ungegn/countrylinks.html; http://www.courageservices.com/

Volunteered Geographic Information is a promising vernacular source of geographic, in contrast to the more authoritative, official sources [12], [9], [14].

Many crowd-shared resources such as Wikipedia, Wikimapia and YouTube include geographical metadata. Souza et al. collected local references for Brazil from a national mapping agency [30], the postal company and some local government institutions, but even that was insufficient, so they included tourist websites. Others have mined places with their geographical extent from web pages [3], or mined places and determined their geographical extent from references such as postal code [32], or through a specially made tool such as the Jeocrowd to search user-generated data sets [19]. Others have asked users to help generate place names by setting up a separate web platform and asking people to add place names directly [31], or asking users to photograph every kilometer of the earth's surface and mine the place name tags indirectly, in the Geograph project[3].

We restrict our attention to resources that include geographical coordinates as well as well as place names, harvesting example tags from Chennai, India from OpenStreetMap and Wikimapia.[4]

Others' approach: gazetteer creation and merge

Beard [2] outlines characteristics of a gazetteer based on Volunteered Geographic Information. Peng et al. [24] describe an architecture for a digital gazetteer that accepts place names from Web 2.0 sources. Kessler, Janowicz, Bishr et al. [18] proposed that the next generation gazetteer be built on a "wiki" model so that all entries are user-generated. Their suggestions for a next generation gazetteer infrastructure, such as the ability to harvest place names in blog posts and the ability to align new data with existing data is appealing. That is in essence the infrastructure we have built, although we prefer to retain the existing gazetteer with core toponyms hand-generated by experts.

Current methods build or augment gazetteers automatically by mining the web [26], [11], [8]. Others have experimented with gazetteer building by using social network sources [17], [20]. Work has even been done in the field of geo-tagged social media [28]. Our aim is not to try to mine a large number of geographical information sources, as did [25]. Nor is the aim of this research to build a comprehensive gazetteer resource, as did [22] and [1]. Our aim is to extract and compare the place names with a gazetteer, as did [19]. We determine the novelty of each entry for the purpose of integration into a gazetteer, as did [17], although the Kessler team did not consider the spatial precision of each extracted geo-tag. Gazetteer enrichment and evaluation work has been performed by [26], [25].

Others' approach: gazetteer merge and fuzzy match

We enrich an existing gazetteer rather than build our own, so as to retain the core, hand-generated, high quality entries. Our research is therefore also in distinguishing between what mined entries are the same and what are different from the current gazetteer. Satisfaction of match between gazetteer entry and toponym has leaned on name spelling, feature type (park, plaza, etc.), and spatial relationship [33] and semantic relations similarity and temporal footprint [23].

Our fuzzy match algorithm uses spatial constraints and allows some semantic ambiguity, as does the lingual location search algorithm of [15]. The JRC Fuzzy gazetteer is fuzzy in semantic search [16]. How to merge the newly-found information with the existing information has been considered by [23], [29], [5]. Examples of merge problems are that the same place may have entirely different names, or different accepted spellings, or spatial relations between places might be unclear [21].

Martins [23] has the same objective of duplicate detection for a gazetteer as we do, and also uses SVM as well as alternating decision trees. His accuracy is quite high. Our data sets are different, however, so comparing our results would be misleading. Our method differs from Martins' in that we automatically generate pairs while Martins has manually generated pairs and classifies them automatically.

2. User geo-coded data

We opted for mining toponyms and coordinates from Web 2.0 mapping applications for the sake of efficiency since tags are clustered in and practicality. OpenStreetMap, which started in London and evaluated in London [13] and found to be within about 6 m of the position recorded by the Ordnance Survey, which is considered to have higher overall quality.

The tags are added to continually, and occasionally updated by others in some applications such as OpenStreetMap, makes them useful, despite the lack of authoritative sponsor. The volume of data may compensate for reliability that we could count on from a more authoritative source, such as a map issued by a federal government. That more people have reviewed the information to add to reliability has been called Linus's Law, after Linus Torvalds, on the principle of open source software [7].

Table 1 compares properties of each Web 2.0 system

	User-supplied		**Authoritative**
	OpenStreetMap	**Wikimapia**	**GeoNames**
Language per entry	single language	single language	multiple languages
Location categories (building, etc.)	yes	yes	yes
Photo	no	optional	optional
Spatial Hierarchy	no	no	yes

We extract place names that have come with tags as have been supplied for OpenStreetMap and Wikimapia. Table 1 compares properties of the two applications. Open Street Map is a platform for contributing and viewing geographic information. Wikimapia is a multilingual, open-content resource where users drop place names with descriptions and proof links and the optional photograph.

Tag quality

We are more assured of tag quality when the tags are duplicated between geographic applications. Table 2 gives example tags from associated with Chennai.

[3] www.geograph.org.uk

[4] We started with Flickr, too, but the Yahoo-owned Flickr uses the geo-database from GeoPlanet rather than raw names entered by users.

Tag geographic coordinates

Schockaert [27] allows that spatial boundaries for some place names may be vague. Tag coordinates from our OpenStreetMap and Wikimapia data might be associated with a spot that is not necessarily that region's geographical center. Moreover, user interfaces from these two applications also introduce imprecision. The input mode for geographic coordinates for OpenStreetMap and Wikimapia is similar in that both provide a map base on which the user marks places.

Table 2 gives examples of tag types in Web 2.0 sources

Wikimapia or OpenStreetMap Tags	Tag characteristics
Madhavaram Taluk, Thiruvallur District - மாதவரம் வட்டம், திருவள்ளூர் மாவட்டம்	Alternate languages
Loyola College	Alternate levels specificity
Loyola College Campus	
Kathipara Flyover or Nehru Circle	Formal and colloquial names
CIT Colony	Shortened Forms
Kumaran Nagar	Name only
Kumaran Nagar 1st Street, GKM colony,CH-600082	Entire address
danny thesis	Noise
Annanagar	Alternate forms (Nagar = neighborhood)
Anna Nagar	

Geographical accuracy

Accuracy is the degree to which information matches true or accepted values. Would a map made with toponyms mined from OpenStreetMap and Wikimapia be accurate? For accepted values, we used Google Maps rather than GeoNames. This was the result of a conversation with Mark Wick, the founder of GeoNames.[5] To verify that GoogleMaps was complete enough for an experiment, we compared a random sample of 100 Chennai tags from OpenStreetMap and Wikimapia to entries in Google Maps.[6] We found that 97% were in Google Maps.

To test for geographical accuracy, we randomly selected 100 valid Chennai tags from OpenStreetMap and Wikimapia and compared their coordinates to those in GoogleMaps and also to GeoNames. Surprisingly, we have that not only does OpenStreetMap have a much higher accuracy than does Wikimapia, it also has a higher accuracy than GeoNames in comparison to the Google Maps geo-coordinates for those same places (Fig. 1).

Are the OpenStreetMap coordinates accurate enough for a city-scale map? At 1:24,000 scale, which is city scale, 1/50th of an inch is 40 feet (12.2 meters), which is considered acceptable accuracy.[7] The Fig. 1 results show that none of the sources, not even OpenStreetMap, have city-scale accuracy if GoogleMaps is considered the gold standard.

An alternative to obtain more precise locations would be to mine addresses with street names, cities and zip codes, as did [11]. However, these addresses will not be rich in vernacular names, as are the sources we use for our study.

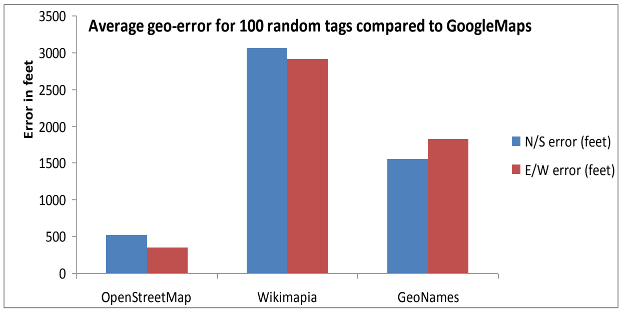

Figure 1 shows the N/S and E/W error in feet for 100 randomly-chosen tags for each of our Web 2.0 sites, and from our GeoNames gazetteer in comparison GoogleMaps

3. METHOD

3.1 Fuzzy-duplicate detect method and data

The objective of the algorithm is to determine duplication among place name, geo-coordinate matches. Our proposed method finds duplicates among the crowd-sourced data first. Duplicates here are confirmation of reliability, and help to reduce noise. Our method then looks for duplicates between crowd-sourced data and the gazetteer to determine which are novel. The algorithm accounts for the expected user-generated variety in spelling and imprecision in geo-coding in determining whether any two entries match.

3.2 Fuzzy duplicate detect architecture

Fig. 2 charts our experimental system for fuzzy duplicate detection. The algorithm uses Lucene to index the gazetteer.[8] We created the features manually, although statistical feature selection methods such as the Lasso regression could be used alternatively.

Before constructing the query, we check the synonym set to see whether similar entries can be made. Then a query is made from the original word form, its bi-gram, tri-gram, and its geo-coordinates. A related query is generated from the word with any synonyms. The weights for the features are determined by a Support Vector Machine process which uses a Support Vector Regression method.[9] This is labeled in Fig 2 as "learning weights for the query".

Two separate post-processing pipelines in Fig 2 are labeled "baseline" and "advanced". The baseline pipeline is language independent. Even for languages in which there is no training data—and so cannot use the advanced pipeline—we will be able to get reasonable results from the baseline.

[5] Mark Wick from GeoNames, email to Gelernter, May 21, 2013.

[6] Chennai, India; Chiclayo, Peru; Alexandria, Egypt

[7] United States Geographical Survey, Map Accuracy Standards Fact Sheet from 1999, from http://egsc.usgs.gov/isb/pubs/factsheets/fs17199.html

[8] http://lucene.apache.org

[9] We used the LibSVM package, in August 2013 at http://www.csie.ntu.edu.tw/~cjllin/libsvm

The output for the baseline is categorized based on the confidence value generated as match (high confidence value), guess match (fairly high confidence), similar (some confidence) and non-match (low confidence). The confidence levels defining the categories were set by experimentation. The output for the advanced SVM method is either a match or not. Those geo-tags which do not match with the gazetteer become novel entries.

The outputs from baseline are “match”, “guess match”, “similar” and “no match (with or without containment)”, with the matches from the baseline in the “similar” category given to a human curator to judge. The matches generated by the Advanced SVM are just match or no match.

3.3 Overview of the procedure

1. We pre-process the data before running the algorithm. Pre-processing consists of changing all characters in the data and also in the gazetteer to lower case, removing the punctuation, and de-accenting the characters. Then we tokenize, and then run the match algorithm over the data. The algorithm originated in a mis-spell algorithm which aims to find the best candidate for a mis-spelled word (described in [10]).
2. We added the ability to consider matches only within a certain geographic range (here, we used a -0.5 and 0.5 latitude and -0.5 and 0.5 longitude). This could be refined later by adding the latitude of the city, and also the population density (so that a rural area might have a wider buffer, for example).
3. We ran the fuzzy match algorithm over the extracted Web 2.0 data, with the GeoNames as the gazetteer lookup.
    a. We generated exact matches if the word is exactly matched with a candidate in the gazetteer.
    b. We generated partial string matches [example: Adambakkam Police Station in the Web 2.0 data is a partial match with Adambakkam in GeoNames]. If the relationship shows one is a part of the other, output as "containment". These contained places do not match with any in the gazetteer.
    c. A manually-generated knowledge base helps reduce the mis-match caused by synonyms that are semantically related by formally different, such as brook and stream. If a word in the entry is contained in the synonym dictionary, we use each synonym word to form different queries for the entry.
    d. We constructed a weighted feature query to search the gazetteer. The weights are generated by training an SVM using the tagged data. (The tagged data consists of string pairs and match or non-match tags). The features used to train the SVM are consistent with the features used to construct the query, which are bigram, trigram and complete word form. Weights are generated from the SVM.
    e. Candidates are generated by selecting the top 10 results generated by the Lucene ranking algorithm, which uses an optimized tf-idf ranking algorithm. We experimented with the top 3 results, but decided to use the top 10 results as candidates to increase recall.
    f. We use two methods to rank the gazetteer candidates (1) edit distance as the baseline method, and (2) candidate re-ranking with SVM.
        - Our baseline method re-ranks match candidates using pure string similarity (edit distance)

$$\frac{1-d\,(\text{edit distance})}{\text{target string length}}$$

Our confidence values are therefore between [0, 1]. The thresholds for the confidence values were determined experimentally, and are somewhat data-dependent. This is necessary because there are too many entries to examine by the classifier. This step acts as a pre-selection for potential matches. The algorithm outputs only those candidates within threshold to reduce the load for manual curation (candidates marked “similar”).

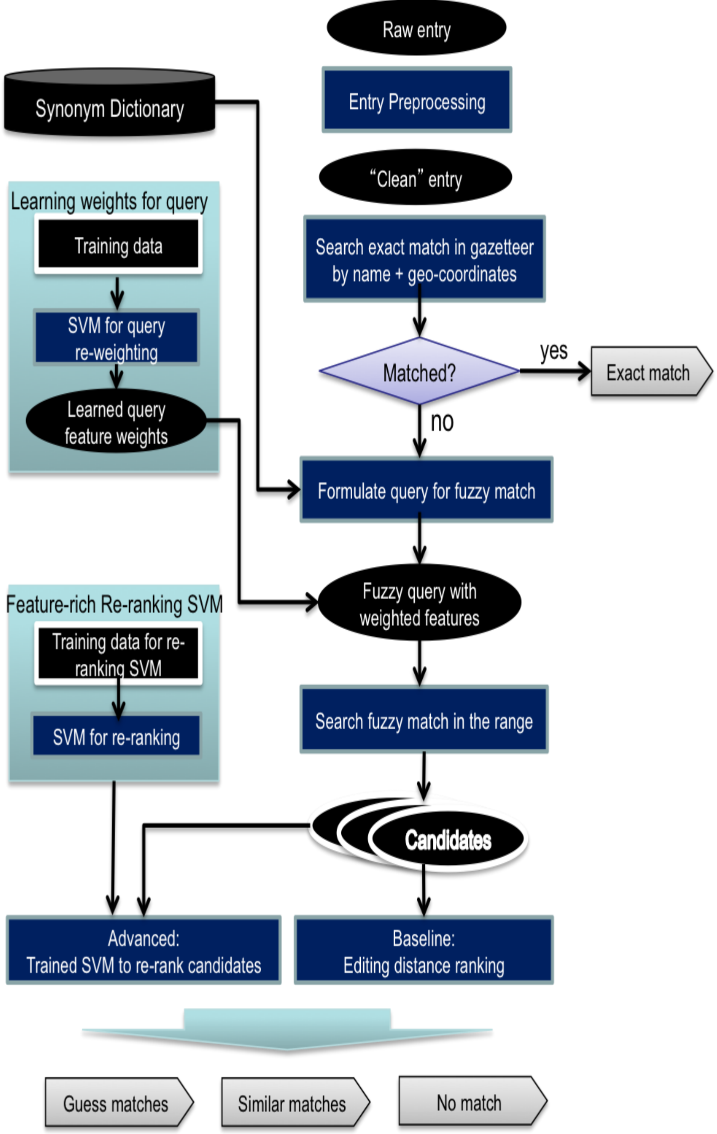
Raw entry
Synonym Dictionary
Entry Preprocessing
“Clean” entry
Learning weights for query
Training data
SVM for query re-weighting
Learned query feature weights
Search exact match in gazetteer by name + geo-coordinates
Matched?
yes
Exact match
no
Formulate query for fuzzy match
Feature-rich Re-ranking SVM
Training data for re-ranking SVM
SVM for re-ranking
Fuzzy query with weighted features
Search fuzzy match in the range
Candidates
Advanced: Trained SVM to re-rank candidates
Baseline: Editing distance ranking
Guess matches
Similar matches
No match

Figure 2. Chart of major steps in our experimental procedure to arrive at a fuzzy match algorithm

- For the SVM candidate re-ranking method, we check whether every query tag and gazetteer candidate produced by the baseline result is an actual match, which is done by classifying each query-candidate pair as match or no-match with a probability given by SVM. We rank the match candidates in descending order, and output only the one with the highest probability.

g. Attach words from the geo-spatial hierarchy (example: province, country) to entries as preparation for inclusion in the gazetteer.
h. Generate a new ID if the entry does not yet exist in GeoNames.

4. Manually determine for the baseline method whether tags in the “Similar” category are matches actually, whereas the SVM does not need this step. [10]
5. For all manually-verified novel entries, add Country, Province/State, City (from bounding box used to extract data). All instances of containment count as new entries if the entry is more specific than an existing gazetteer candidate.

3.4 SVM for query weighting and candidate re-ranking

Both the query weighting step and candidate re-ranking step require SVM training. These steps correspond to the two left most sub-process in Figure 2. The query weighting step uses SVM to help determine the weights of the query features (see section 3.3 step 3d), whereas candidate re-ranking uses a wider feature set to re-rank the candidates using classification.

SVM Features for query weighting.

The query is the extracted geo-tag. The features for query weighting reflect characteristics of those geo-tags:

F_1 = word-level similarity. This represents the number of the words that are matched in both the geo-tag and the candidate, divided by the number of words in the geo-tag.

F_2 = The proportion of the matched bigrams divided by the number of bigrams in the geo-tag.

F_3 = The proportion of the matched trigrams divided by the number of trigrams in the geo-tag.

Each of these features aggregates numerous terms in the feature vector for the query. To determine weights for the terms, we borrowed the notion of “long query problem” from ad hoc information retrieval. One of the ways to solve the problem is to learn the importance of the term for a specific query term vector. However, in our problem, we want to figure out the weight for each group of terms instead of for a specific single words, bigram or trigram. So we learn through the SVM the weight of F1 F2 and F3, which helps determine the weights for words, bigrams, and trigrams.

The training data includes 160 location phrases selected randomly from the Chennai OpenStreetMap and Wikimapia data. We used the pipeline on the right-hand side of Figure 2 (without SVM) to generate the candidates and manually tag each pair as match or not. We train the SVM regression model on this data to figure out the coefficients for **F_1, F_2 , F_3.**

Table 3: Weights learned with SVM Regression in LibSVM, with linear kernel C = 0.1.

Feature	coefficient
F1	0.124
F2	0.607
F3	-0.614

The coefficients are used to generate weights in the query (See Table 3). The coefficients show that the bigram feature is most helpful in finding a match between candidate term and gazetteer term. The bigram works better than the word or trigram matching because it is more tolerant of spelling errors, while it preserves the character order.

SVM for candidate re-ranking

We use the query weighting features along with these additional features to train the SVM model for re-ranking gazetteer candidates generated from step (f) in section 3.3.

F_1 , F_2 , F_3 as above

F_4 = head matching proportion of geo-tag and candidate

Treats matching as a string between letters at the beginning of the geo-tag and beginning of the gazetteer candidate, and finds the longest string match counting from the first character.

F_5 = average head-matching proportion of geo-tag and candidate

Here we tokenize a geo-tag phrase into words first, and then for each word, we find the longest head-matching length among all the words in the candidate, and normalize with word length. Finally we take the average among the normalized head-matching scores.

F_6 = normalized geographical distance between geo-tag and candidate.

$$Rel_{distance} = \frac{1- \text{ distance (geo} - \text{tag, candidate)}}{(0.5 * 2^{0.5})}$$

Then the relevant distance is in the range [0,1]. The reason why we use 0.5 * $2^{0.5}$ is because we use a bounding box as the geographic range, so the longest possible distance between geo-tag and candidate is not 0.5, but each corner point of the square.

F_7 = edit distance between geo-tag and candidate, which is identical to the baseline measurement

F_8 = containment, where one entry is contained within another entry

F_9 = Soundex code match

Soundex is an algorithm that uses phonetics (sound in speech) to aid in matching. F9 is used to address the vowel mismatch problem. In Indian English, lots of the places use “oo” instead of “u”, and “th” instead of “tt”. We use soundex to map those into the same numerical value, to find the equivalence of the sounds.

3.5 Evaluation

For the baseline method, we randomly selected 200 geo-tags from OpenStreetMap and Wikimapia. Each entry consists of a location word or phrase plus latitude and longitude, presumably of its centroid. There are four types of output from the baseline: Match, Guess match, Similar, No match. Examples of each are in Table 4, as are the features that we inferred from these and other instances. The algorithm finds a gazetteer match for every tag. The "no match" decision is based on a poor gazetteer match with a given tag.

We calculated precision and recall for the baseline method for 200 randomly-selected tags from the OpenStreetMap and Wikimapia Chennai data. If the fuzzy match algorithm found a match with the GeoNames that it should have found, we considered precision to be correct. We did not count instances of “guess match” and “similar” in our evaluation because there will be a person to judge

[10] This probability for baseline match here "Match" is an exact match, and “Guess match” is >85% confidence, "Similar" 70% < x < 85%, "No match" is x<70%.

these. For the baseline, our precision = .920, recall .830, and F1=.875.

From this 200-tag sample, 88.5% are non-matches, that is, novel entries to be added to the gazetteer. Of these, 14% represent containment (a subset of entries already in the gazetteer), and 74.5% are entirely novel.

Table 4 examples from OpenStreetMap and Wikimapia, the gazetteer candidate, and the baseline judgment.

Crowd-sourced tag	**GeoNames gazetteer entry found**	**Features**	**Baseline decision**
Perambur 13.10, 80.24	Perambur 13.11 80.24	Direct match.	Match
Pulianthope 13.09, 80.26	Puliantope 13.10, 80.26	Editing distance, soundex	Guess Match
Purusawalkam 13.08, 80.25	Purasawalkam 13.08, 80.25	Editing distance, soundex	Guess Match
Perambur Loco Works 13.10, 80.22	Perambur Locoworks 13.10, 80.22	Editing distance, average head matching	Similar
Perungalathur 12.90, 80.09	Perunkalattu 12.91, 80.08	Head matching, soundex	Similar
Pudupakkam 13.24, 80.21	Madipakkam 12.97, 80.20	Head matching	No match
Agaram Mel 13.03, 80.07	Agaram 13.03, 80.08	Average Head matching	No match, Containment

For the Advanced SVM classification method, we used 1768 geo-tag/candidate pairs, which contains 65 containment (these will be no matches), 69 matches, and 1634 no matches. Table 5 shows that the features outlined in section 3.4 are effectively encoded in SVM with the RBF kernel.

Table 5: SVM with 9 features for gazetteer candidate re-ranking using a linear kernel vs RBF kernel

Kernel		Match	Containment	No match	Overall
linear	P	0.899	0.908	0.990	0.984
	R	**0.886**	0.855	0.993	0.984
	F1	0.892	0.881	0.992	0.984
RBF	P	**0.952**	**0.913**	**0.990**	0.988
	R	0.843	**0.984**	**0.998**	0.988
	F1	**0.894**	**0.947**	**0.994**	**0.988**

We can see in Table 5 that the RBF kernel surpasses the linear kernel in the overall F1 statistic, however, the linear kernel gives a higher recall. This recall is important because if there is a duplicate in the gazetteer, we do not want to add a separate, repetitive entry.

The containment category is high enough if we use RBF kernel to substitute for the high accuracy heuristics in practical system. For the no match category, the accuracy is high because unmatched training pairs dominate the training data.

We compared different feature combinations for the candidate re-ranking SVM in Table 6 to test the effectiveness of the separate features.

Table 6: Precision, recall and F1-statistic for feature combinations for candidate re-ranking SVM (RBF kernel). Bold numbers indicate the highest precision, recall and F1 per column.

Features		Match	Contain-ment	No Match	Overall
1,2,3	P	0.943	0	0.940	0.940
	R	0.471	0	1	0.940
	F1	0.629	0	0.970	0.940
1,2,3,8	P	0.943	0.877	**0.992**	0.968
	R	0.471	0.826	0.972	0.968
	F1	0.629	0.851	0.995	0.968
1,2,3,8, 9	P	0.843	0.879	0.988	0.978
	R	0.843	0.841	0.990	0.978
	F1	0.843	0.860	0.989	0.978
1,2,3,7, 8,9	P	0.932	**0.970**	0.987	0.985
	R	0.786	0.913	0.996	0.985
	F1	0.853	0.940	0.992	0.985
1,2,3,6, 8,9	P	0.855	0.879	0.988	0.980
	R	0.843	0.841	0.991	0.980
	F1	0.849	0.860	0.990	0.980
1,2,3,4, 5,8,9	P	0.850	0.910	0.990	0.981
	R	**0.886**	0.841	0.991	0.981
	F1	0.867	0.872	0.991	0.981
All (1-9)	P	**0.952**	0.913	0.990	**0.988**
	R	0.843	**0.984**	**0.998**	**0.988**
	F1	**0.894**	**0.947**	**0.994**	**0.988**

Table 6 shows that the features outlined in section 3.4, as added one by one, improve our overall system performance, as shown in the F1 value. First, we used the features for the query-weighting only. Although the overall accuracy is high, there is not enough information to accurately distinguish the match from the no match. What makes things worse is that containment is not recognized. In order to address this problem, we added containment feature F8 to the model, which greatly boosted the accuracy for containment. F9 soundex feature is used to alleviate the mis-match problem introduced by transliteration error. Next we added edit distance feature F7, and the score did not seem to improve much. The same was the case for distance feature F6. We tried the head matching features F4 and F5 which boosted the score a lot for the matches and a little bit for containment and no matches. Finally, we add all the features together, and got the highest F1 statistic.

To conclude, using all the features combined, our candidate re-ranking SVM achieves higher F1 statistic than the edit distance baseline. The SVM method alone could be used as an automatic gazetteer expansion method.

4. DISCUSSION

The number of novel matches between the extracted geo-tag set and the gazetteer demonstrate the utility of Volunteered Geographic Information for gazetteer enrichment, our research question 1. Our scores demonstrate the effectiveness of our feature set and the SVM with an RBF kernel in declaring whether a geo-tag is a match with the gazetteer, research question 2.

Our algorithm uses two features which we have not found in similar research. We use the head matching, average head matching, and Soundex features. These help us to increase our F1, as shown in Table 6. Our nine feature advance method to automatically generate potential gazetteer matches allows the system to generate candidates of potentially higher relevance, so there is a higher possibility that we will find the correct match. This serves to increase recall.

Preliminary experiments with Arabic geo-tags using our baseline method gave results that are acceptable. This is because the baseline rests upon editing distance, and editing distance uses string similarity which is language independent.

Our evaluation could be reproduced by downloading a set of geo-tags (location name + geo-coordinates) for Chennai or another city from either or both of OpenStreetMap and Wikimapia, and running the tags through our fuzzy match algorithm.[11]

5. FUTURE ALGORITHM RESEARCH

Scale-up of this procedure would require automatic downloading of the geo-tags from geocrowd sources. For a range of languages, we could use the baseline method and give the "similar" results to a person to decide whether the pair constitute a match. Or we could use the advanced method with SVM, which would require training data and a Soundex implementation for each language.

Many steps performed manually for these experiments could be automated in future. Here, we manually removed noisy tags (see "danny thesis" tag in Table 2, for example) for these experiments. In the future, geo-tags that have no obvious match could be sent to Google Maps first for preliminary filtering for quality. And for data that must be sent to a person for final judgment, we could make the curation task faster and easier by visualizing geo-tags on a map, so that a person can verify the geo-coding to some extent, and make changes to the crowd-sourced data if required. To determine which matches generated by the advanced method might be uncertain, we will add a confidence value.

We used a small knowledge base to find semantic parallels (brook and creek, for example.) We anticipate that a wider knowledge base would improve recall further. An English-language geospatial ontology would improve place name matching when a feature is part of a name, and facilitate feature matching. The GeoNames feature codes are not extensive,[12] nor are inter-relations among the features provided (such as the fact that a hill is a small mountain). Both features and training data for a machine learning algorithm would be easy to extract from Open Street Map. And we would have training data for many languages. Wikipedia and OpenStreetMap include tag categories (see Table 1) so that we can know whether the geo-coordinates represent a spatial point or a region centroid.

The GeoNames gazetteer has an RDF representation which opens options for semantic linking and obtaining additional attribute and alternate geometry information. Research would entail attaching polygons to toponyms so that the new entries could be used in Geographic Information Systems.

6. SUMMARY

The gazetteer is a core of geographic information retrieval, and having more entries will serve to improve recall, as long as precision does not suffer with more false positives. Collecting smaller-scale location names, city by city, all over the world is time-consuming, and so automatic means for gazetteer enrichment are critical. This paper experiments with both potential sources of crowd-sourced geo-tags and a method to determine whether each tag is a duplicate or novel for the gazetteer.

Of OpenStreetMap and Wikimapia, the OpenStreetMap tags were more geographically accurate and plentiful for our sample city, Chennai, India. If we consider GoogleMaps a standard for relative geographical accuracy, however, even the OpenStreetMap places are not precise enough for city-scale mapping for most applications.

These downloaded, crowd-sourced Chennai geo-tags became data for our fuzzy match algorithm to determine whether the places were already in the gazetteer or whether they would be new entries. We found that 88.5% of the tags were novel, suggesting that OpenStreetMap and Wikimapia would make good sources for gazetteer enrichment. Tests should be expanded to many other cities.

Our fuzzy match duplicate detect algorithm has both a language-independent baseline method with an F1 of .875 and a better performing machine learning pipeline with an F1 of .894 for the same geo-data. The machine learning pipeline, however, requires training data and a Soundex implementation for the language of the place names. Even so, results demonstrate that our fuzzy match algorithm could be considered fundamental infrastructure for automatic gazetteer enrichment.

7. ACKNOWLEDGMENTS

We acknowledge and thank OpenStreetMap, and Wikimapia users for the opportunity to work with their data.

8. REFERENCES

[1] Axelrod, A.E. 2003. On building a high performance gazetteer database. Eds. K. and B. Sundheim, *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, 63–68.

[2] Beard, K. 2012. A semantic web based gazetteer model for VGI. *ACM SigSpatial GeoCrowd '12, November 6, 2012, Redondo Beach, CA*, 54–61.

[3] Blessing, A. and Schütz, H. 2008. Automatic acquisition of vernacular places. *Proceedings of the 10th international conference on information integration and web-based applications and services, iiWAS2008, November 24-26, Linz, Austria*, 662–665.

[4] Carroll, A. (2006). A Case Study In Progress: How a Media Organization Tackles the Georeferencing Challenge /Opportunity. http://ncgia.ucsb.edu/projects/nga/docs/carroll-paper.pdf

[5] Cheng, G., Lu, X, Ge, X. Yu, H., Wang, Y. and Ge, X. 2010. Data fusion method for digital gazetteer. *18th International Conference Geoinformatics, 18-20 June 2010, Beijing, China*, 1-4

[11] As of October 7, 2013 at https://github.com/geoparser/geolocator/tree/master/geo-locator/src/edu/cmu/geoparser/nlp/spelling. Fuzzy match file in the package is called LearningToRankDictionaryMatching.java

[12] http://www.geonames.org/export/codes.html

[6] Chiang, Y-Y. 2010. *Harvesting geographic features from heterogeneous raster maps.* A PhD dissertation. University of Southern California, December 2010.

[7] Elwood, S., Goodchild, M. F., and Sui, D. Z. 2012. Researching volunteered geographic information: Spatial data, geographic research, and new social practice. *Annals of the Association of American Geographers, 102* (3), 571–590.

[8] Furche, T. Grasso, G., Orsi, G., Schallhart, C. Wang, C. 2012. Automatically Learning Gazetteers from the Deep Web. *WWW'12 Apr 16-20, 2012 Lyon, France,* [4 pp.]

[9] Gelernter, J. 2009. "Neogeography" in *Handbook of Research on Social Interaction Technologies and Collaboration Software.* (Eds. T. Dumova and R. Fiordo).

[10] Gelernter, J. and Zhang, W. 2013. Cross-lingual geo-parsing for non-structured data. 7th *Workshop on Geographic Information Retrieval (GIR) November 5, 2013, Orlando, Florida, U.S.A.*

[11] Goldberg, D.W., John P. Wilson & Craig A. Knoblock 2009. Extracting geographic features from the Internet to automatically build detailed regional gazetteers. *International Journal of Geographical Information Science* 23 (1), 93 –128.

[12] Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal 69*(4), 211–221.

[13] Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning & Design 37*(4): 682-703.

[14] Ho, S. and Rajabifard, A. 2010. Learning From the Crowd: The Role of Volunteered Geographic Information in Realising a Spatially Enabled Society. *GSDI 12 World Conference: Realising Spatially Enabled Societies, 19 –22 October, Singapore.*

[15] Joshi, T., Joy, J., Kellner, T., Khurana, U., Kumaran, A., Sengar, V. 2008. Crosslingual location search. *SIGIR '08, July 20-24, 2008, Singapore*, 211–218.

[16] JRC Fuzzy Gazetteer. From Joint Research Centre of the European Commission. See http://dma.jrc.it/services/fuzzyg/

[17] Kessler, C., Maue, P, Heuer, J.T., Bartoschek, T. 2009. Bottom-up gazetteers: learning from the implicit semantics of geotags. K. Janowicz, M. Raubal, and S. Levashkin (Eds.) *GeoS 2009, LNCS 5892*, 83–102. Springer, Heidelberg.

[18] Kessler, C., Janowicz, K., Bishr, M. 2009. An agenda for the next generation gazetteer: geographic information contribution and retrieval. *ACM GIS '09, November 4-6, 2009, Seattle, WA*, 91–100.

[19] Lamprianidis, G., and Pfoser, D. 2012. Collaborative geospatial feature search. *ACM SigSpatial GIS, November 6-9, 2012, Redondo Beach, CA,* [10 p.]

[20] O'Hare, N. and Murdock, V. 2013. Modeling locations with social media. *Information Retrieval 16* (2013), 30–62.

[21] Machado, I. M. R., de Alentar, R.O., de Oliveira Campos Jr., R., Davis Jr., C.A. 2011. An ontological gazetteer and its application for place name disambiguation in text. *Journal of the Brazilian Computer Society. 17*(4), 267-279.

[22] Manguinhas, H., Martins, B. and Borbinha, J. 2008. A geo-temporal web gazetteer integrating data from multiple sources. *Third International Conference on Digital Information Management, ICDIM 2008. 13-16 Nov. 2008,* 146–153.

[23] Martins, B. 2011. Supervised Machine Learning Approach for Duplicate Detection over Gazetteer Records. *GeoSpatial Semantics Lecture Notes in Computer Science 6631* (2011), 34 –51.

[24] Peng, X. 2010 A folksonomy-ontology-based digital gazetteer service. *18th International Conference on Geoinformatics, 18-20 June 2010, Beijing, China*, 1–6

[25] Popescu, A., Grefenstette, G. Bouamor, H. 2009. Mining a multilingual geographical gazetteer from the web. *2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology –Workshops*, 58-65.

[26] Popescu, A., Grefenstette, G., Moellic, P.A. 2008. Gazetiki: automatic creation of a geographical gazetteer. *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries*, 85-93.

[27] Schockaert, S. 2011. Vague regions in geographic information retrieval. *SigSpatial Special 3*(2), 24–28.

[28] Sizov, S. 2012 Latent geospatial semantics of social media. *ACM Transactions on Intelligent Systems and Technology,* 3, 4 (2012), 64:1 –64.20

[29] Smart, P.B. Jones, C.B. Twaroch, F.A. 2010. Multi-source toponym integration. *Proceedings of the 6th International Conference on Geographic Information Science, GIScience'10* Heidelberg: Springer, 234 –248.

[30] Souza, L.A., Davis Jr., C.A., Borges, K. A.V., Delboni, T.M., Laender, A. H. F. 2005. The role of gazetteers in geographic knowledge discovery on the web. *Proceedings of the Third Latin American Web Congress LA-WEB '05*, 1-9

[31] Twaroch, F.A. and Jones, C. B. et al. 2010. A web platform for the evaluation of vernacular place names in automatically constructed gazetteers. *GIR '10 Proceedings of the 6th Workshop on Geographic Information Retrieval*. [2 p.]

[32] Twaroch, F.A., Jones, C.B. and Abdelmoty, A.I. 2009. Acquisition of vernacular place names from web sources. I. King, R. Baeza-Yates (eds). *Weaving Services and People on the World Wide Web*, 195–212.

[33] Yu, H., Wang, X., Chen, M., Li, R., Chen, K. 2010. Calculating of match degree in digital gazetteer services. *18th International Conference on Geoinformatics, 18-20 June, 2010, Beijing, Chin*a, 1–6.