An Efficient Location Extraction Algorithm by Leveraging Web Contextual Information

Teng Qin
Peking University
No.5 Yiheyuan Road, Beijing, China
palmtenor@gmail.com

Rong Xiao
Microsoft Research Asia
Sigma Building, 49 Zhichun Road, Beijing, China
rxiao@microsoft.com

Lei Fang
Tsinghua University
No. 30 Shuangqing Road, Beijing, China
jsfanglei@gmail.com

Xing Xie
Microsoft Research Asia
Sigma Building, 49 Zhichun Road, Beijing, China
xingx@microsoft.com

Lei Zhang
Microsoft Research Asia
Sigma Building, 49 Zhichun Road, Beijing, China
leizhang@microsoft.com

ABSTRACT

A typical location extraction approach consists of two steps, location name detection and location entity disambiguation. Promising results have been obtained in the last decade based on natural language processing technologies. However, there are still two challenges which requires further investigation: 1)How to leverage the prior and contextual evidence to improve the location extraction performance, and 2) How to utilize the interdependence information between the named entity recognition step and disambiguation step. In this paper, we propose an iterative detection-ranking framework to address these problems as well as a set of novel features to mine contextual information from web resources. Experimental results show that our solution outperforms the state-of-the-art approaches, including Metacarta GeoTagger and Yahoo Placemaker.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining, Spatial databases and GIS*

General Terms

Algorithms, Performance, Experimentation

1. INTRODUCTION

In recent years, location-based services/applications have undergone rapid growth. According to the comScore study, local search grew 58 percent in 2008, significantly outpacing the 21 percent growth in overall U.S. core Web search during the same period. Numbers from research firm Gartner also show that the soaring use of location-based services will lead to a doubling of subscribers and revenues by the end of 2009. General web pages hold a huge amount of geographic information. However, only a minute proportion of this data is accompanied with machine readable meta-data. Therefore, how to correctly and effectively detect geographic locations from web resources has become a key challenge for location-based services/applications.

As mentioned in [5], location extraction usually solves two kinds of ambiguities, geo/non-geo and geo/geo. A geo/non-geo ambiguity occurs when a place name also has a non-geographic meaning, such as a person name "Washington" or a common word "Turkey". Geo/geo ambiguity arises when distinct places have the same name, for example, there are 23 cities named "Buffalo" in the U.S.

In order to solve these ambiguities, a location extraction procedure usually is decomposed into location name detection and location entity disambiguation steps either implicitly or explicitly. In the location name detection step, the geo/non-geo ambiguity is solved by identifying the geographic names from common words which can also be regarded as a special case of named entity recognition (NER) problem. In the location entity disambiguation step, the geo/geo ambiguity is solved by assigning the most preferable geographic location to each extracted geographic name. Using this strategy, some promising results are shown in [19, 16, 12].

Traditional location extraction algorithms are based on the information extracted from text context and gazetteers. Although gazetteers hold a lot of information about locations, such as official and alternative placenames, vertical topology and geographic coordinates, gazetteers do not provide any more details on how they are used in context. For example, "Gary's fortunes have risen and fallen with those of the steel industry". In this sentence, we cannot identify whether Gary is a person name or a location name. It makes the task of geographic location extract very challenging. Human beings have a remarkable ability identify correct geographic references from ambiguous and under-specified text using real-world knowledge and experience. Usually two kinds of real-world knowledge could be mined from experience. One is the location reference prior knowledge. Given the word "Paris", most likely it is the capital of France, not the name of Paris city in Illinois, US. An-

ACM GIS '10, November 2-5, 2010. San Jose, CA, USA

other kind of prior knowledge is structural dependencies. For example, given the location "Washington" and the context word "Wizards", most likely it is Washington D.C. not Washington state, since Washington Wizards is a famous baseball team.

Therefore, mining location prior and contextual prior is the key to improve the performance of location extraction. In [19], population data in the gazetteer is used as the location prior. A place with a high population is more likely to be mentioned than a place with a lower one. They also observed that nearby locations appeared in the context will also provide discriminative information for the task of named entity disambiguation. In [16], Overell and Ruger proposed a co-occurrence model to capture the context placenames are used in an extensive set of location synonyms collected from Wikipedia. The experimental results are promising. In [21], besides location population prior, they also propose an algorithm to disambiguate geographical names based on an ontology learned from gazetteers and WordNet.

Due to coverage limitation of WorkNet and Wikipedia documents, these methods are less scalable for large gazetteer. Moreover, the step-wise approach ignores the interdependence between location name extraction or the problem location entity disambiguation. For example, given the word "Gary", most likely it is a person name, not the name of Gary city in Indiana, US. Nonetheless, if we detect a geographic location "Chicago, US", in the context, the probability of the word "Gary" being a geographic location will be significantly increased, since Gary is a nearby City of Chicago, IL. Therefore, in this paper, we argue that the two steps are bidirectional interdependent instead of one-way dependent, i.e. the output of location entity disambiguation may also provide discriminative contextual information for location name extraction. However, due to the limitation of previous step-wise approaches, such information is discarded.

A location extraction procedure can be also considered as a ranking problem, i.e. given an input document, we generate a ranked list of location sense for each candidate term (if we consider non-location as a special location sense). Using this method, the location extraction problem can be formulated as a global ranking problem [18]. However, this global ranking algorithm does not tackle the ranking problem directly but more in the sense of regression. In this section, we proposed an alternative approach to solve location extraction problem jointly. In this algorithm, 1) a location evidence set is proposed to provide geographic contextual evidence for both location name detection and location entity disambiguation, and 2) the location evidence set is iteratively updated with high confidence location items during the evaluation procedure. Moreover, a set novel features are proposed based on the prior and contextual prior knowledge mined from Web.

1.1 Related work

Named Entity Recognition (NER) [17] is a well-known field of Natural Language Processing (NLP) and has been studied over decades. Traditional NER approaches employ machine learning to recognize names from their local structure. In [9], Finkel and et al. propose a novel models of non-local structure with Gibbs sampling. They observed 9% error reduction over state-of-the-art systems. In [5], Amitay et al. develop a Web-a-Where system for associating geography with web pages. They apply two different minimality heuristics to resolve ambiguous location names. The first heuristic used is "one sense per discourse" and the second is that location names mentioned trend to indicate nearby locations. Their method consists of three steps: (1) extract location names mentioned in the given page by using a gazetteer, (2) use four heuristic rules sequentially to disambiguate each extracted location name, (3) use a simple propagation algorithm to compute the focus of the given page.

In recent years, named entity disambiguation has attracted much attention. In [12, 13], Li et al. use a gazetteer to verify geographical names, and proposed a hybrid approach to distill the correct sense of a location name. This method combines (1) linguistic patterns extracted from local context,(2) maximum spanning tree search for discourse analysis, and (3) integration of default senses. Wang et al. explicitly distinguish three types of locations for web resources into three categories: provider location, content location and serving location in [22]. They develop a set of algorithms to compute these categories of web locations based on their specific characteristics. In [8], Ding et al. propose the CGS/EGS algorithm based on geographic content and context sources. In this approach, the authors first define two key measures, namely power for measuring interest and spread for measuring uniformity, and then point out that the geographic scope of a web resource must satisfy enough spread and power. Rauch et al. describe a confidence-based framework for disambiguating location names [19]. They learn the confidence that a location name refers to any geographic location and the confidence that a location name have a special location sense iteratively in a large corpus and disambiguating location names in a given document. This technology is used in Metacarta geotagging service, the leading geographic location extraction service.

Most of these methods can be categorized into two groups rule-based methods and data-driven methods [15]. The rule-based methods [19, 13, 6] are based on manual craft heuristic rules to encode human knowledge into location disambiguation. However, these methods usually suffer from low coverage problem. The data-driven methods generally apply standard statistical learning methods, e.g. Support Vector Machine, to solve the problem of mapping location names to locations. These methods usually require a large accurate corpus with annotation. However, due to the labeling cost, such corpus does not exist in the public domain. Most data-driven approaches are either using supervised learning methods on small sets of ground truth to small domains [14, 11]. Some semi-supervised approaches are proposed to address the problem of lacking annotated data. In these approaches, unlabeled corpus is used in conjunction with a small amount of labeled data can produce considerable accuracy improvement [20]. In [7], Bunescu and Pasca employ several of the disambiguation resources (Wikipedia entity pages, redirection pages, categories, and hyperlinks) and build a context article cosine similarity model and a Rank-SVM based on a taxonomy kernel.

In [23], Wang et al. propose an algorithm to address this problem. They observe that the top search results from search engines are usually relevant and up-to-date. Based on this observation, they mine the top search results and query logs to discover implicit query locations and achieve consistent high accuracy. However, in this approach they only use the location correlation and discard the term cor-

relation information.

2. GEOGRAPHIC KNOWLEDGE MINING

Given a document $d \in D$, each term in d can be represented by a pair $\{w_i, \phi_i\}$, where $w_i \in N$ is the term, ϕ_i is the position of the term in document d. Here L is the list of all locations and N is the list of all geographic location name acquired from a gazetteer.

Due to the geographic name alias/abbreviation, a location $l_i \in L$ may have several names, we use $N(l_i)$ to represent the list of all possible names for a location l_i and use $N_j(l_i)$ to represent the j-th names of the list $N(l_i)$. Moreover, different locations may have the same name, we use $L(n_i)$ to represent the list of all possible location sense for a geographic name n_i and use $L_j(n_i)$ to represent the j-th location sense of the list $L(n_i)$. The task of location extraction is to assign a most possible location sense $y_i \in L$ to each term w_i in the document d. This is a many to many mapping.

There are two kinds of location priors, one is the location prior probability $\mathbb{P}_L(w) = \mathbb{P}(w \in L)$ which means the prior probability of the word w being a location, another is the location sense prior probability $\mathbb{P}(L_j(w))$ which means the prior probability of the location term w have the specific location sense $L_j(w)$. This information is valuable for location disambiguation when the context information is insufficient. For example, we know that $\mathbb{P}_L(\texttt{New York}) > \mathbb{P}_L(\texttt{Flamingo})$ and $\mathbb{P}(\texttt{California, US}) > \mathbb{P}(\texttt{California, ML, US})$.

Usually, human beings can also inference the correct location sense from the location context concluded from the real-world. There are two kinds of location context evidence, one is the context location prior which means the place names appeared in the context tend to indicate nearby locations, another is the context word prior which means terms appeared in the location context tend to be relevant to the location. In the next sections, we will propose a set of algorithms to mine this prior information from WEB.

2.1 Inference Context Location Prior from Web Pages

Usually, an input document may contain several location items. There are two widely used assumptions, 1) the place names appearing in one context tend to indicate nearby locations, and 2) If an ambiguous term appeared several times in a single document, most likely they have the same location sense. These assumptions are usually true for the most documents. Based on these assumptions we proposed a novel algorithm based on score propagation to inference the location correlation knowledge from a Web Corpus.

Suppose we have extracted location items $w_1, \cdots, w_k$ from a given document. Each item w_i may have several location sense $L_j(w_i)$. Suppose, we have observed a list of items, `Pearl Harbor`, `Puhimau Crater`, `Hawaii`. Based on the hierarchy information from the gazetteer, we have following possible location senses, `Hawaii/Honolulu/Pearl Harbor`, `Alaska/Juneau/Pearl Harbor`, `Tennessee/Meigs/Pearl Harbor`, `Hawaii/Hawaii/Puhimau Crater`, `Hawaii`, and `Hawaii/Hawaii`. These location senses can be represented in a tree, as shown in Fig. 1. In this figure, we can find that some location senses, like `Hawaii/Honolulu/Pearl Harbor`, are appeared in the document, and some location senses, like `Hawaii/Honolulu`, are hidden. We use black and gray color to distinguish these two kinds of location senses respectively.

If we want to predict the intended location sense of a name, for example `Pearl Harbor` in Fig. 1. There are three possible location senses in this figure. It is very hard to predict the correct geographic location sense only with this single word. But if we know `Puhimau Crater` also occurred in the article and it is more near to the one in `Hawaii/Honolulu`. We may predict the location sense of `Puhimau Crater` more accurately. Based on this intuition, we proposed a score propagation algorithm. Each appeared location sense $L_j(w_i)$ (nodes in black) has initial score and it can propagate its initial score to other location senses. The initial score is defined by

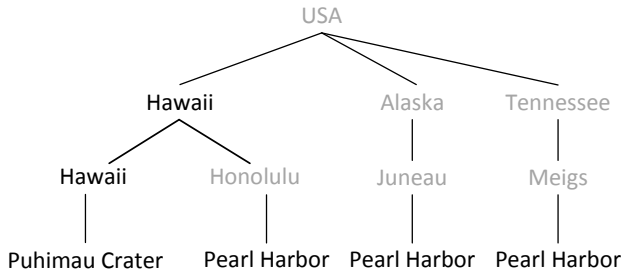

Figure 1: location senses in the location hierarchy tree

$$s(L_j(w_i)|d_t) = \mathbb{P}(L_j(w_i))\mathbb{P}_L(w_i)c(w_i|d_t), \quad (1)$$

where $c(w_i|d_t)$ is the count of geographic name w_i appeared in the given document d_t.

The propagated score will decay as it travels along the tree. Suppose location sense $L_j(w_i)$ will propagate its score. The location senses close to $L_j(w_i)$ will get more score than the location sense far from $L_j(w_i)$. The procedure could be defined by

$$s(L_j(w_i) \to L_q(w_p)|d_t) = \mathbb{P}(L_q(w_p))s(L_j(w_i)) \cdot \alpha^{d(L_j(w_i), L_q(w_p))},$$

where α is the decay parameter and $d(L_j(w_i), L_q(w_p))$ is the distance between location sense $L_j(w_i)$ and $L_q(w_p)$ in the location hierarchy tree. For example, $d(\texttt{Hawaii, US}, \texttt{Hawaii, US}) = 0$, $d(\texttt{Hawaii, US}, \texttt{Pearl Harbor, HI, US}) = 2$.

Sum over scores propagated from different nodes, we have

$$S(L_j(w_i)|d_t) = \sum_{L_q(w_p) \in o_t} s(L_j(w_i) \to L_q(w_p)|d_t), \quad (2)$$

where o_t is the observed location senses in the document d_t. The sum of propagation score could be regarded as the confidence of the location name w_i have the location sense $L_j(w_i)$, which means

$$S(L_j(w_i)|d_t) \propto \mathbb{P}(L_j(w_i)|d_t)\mathbb{P}_L(w_i|d_t). \quad (3)$$

Therefore, after the propagation process, the unique location sense of a location name in the given document d is one of the location senses of the location name that gets the highest score

$$\hat{L}_j(w_i) = \text{argmax}_j s(L_j(w_i)|d_t). \quad (4)$$

Fig. 2 is a demonstration of the score propagation algorithm. In the example shown in Fig. 1, `Hawaii/Hawaii/Pearl Harbor` will be the final location sense for the location name `Pearl Harbor` after the score propagation procedure.

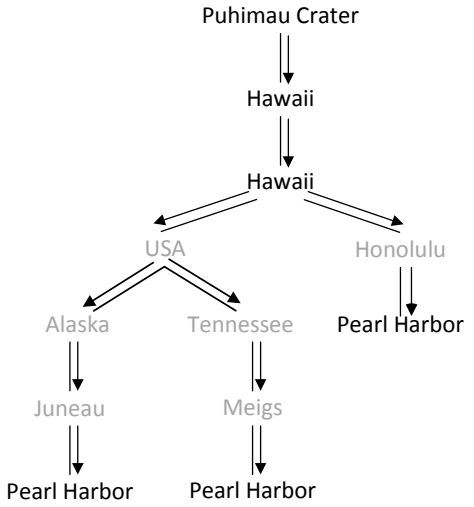

Figure 2: Score Propagation

The goal of the score propagation algorithm is to inference probability $\mathbb{P}(L_j(w_i)|d_t)\mathbb{P}_L(w_i|d_t)$ based on the location context in the document d_t and prior probability $\mathbb{P}(L_j(w_i)$ and $\mathbb{P}_L(w_i)$. Therefore, some heuristic rules can be integrated into this propagation algorithm by adjusting $\mathbb{P}(L_j(w_i)$ and $\mathbb{P}_L(w_i)$ according to document context. For example, if some pattern like `the City of X` appears, then $\mathbb{P}_L(\texttt{X}) = 1$ in the given document. If some pattern like `Central Park, New York, NY` appears,the sense of `Central Park, NY` is clear, other location senses of Central Park will be removed from the tree before propagation.

However, how to evaluate the location priors $\mathbb{P}(L_j(x))$ and $\mathbb{P}_L(x)$ is still unsolved problem. We will discuss this issue in the following section.

2.2 Inference Location Priors from Web Pages

The GeoTag service on Flickr also contains rich information of location prior $\mathbb{P}(L_k(x))$. For each location name x, we submit a query to GeoTag services, and get all possible location senses $L_k(x)$, then another query is submitted to get the number of pictures $n(L_k(x))$ associated with location sense $L_k(x)$. The $p(L_k(x))$ can be estimated by

$$\mathbb{P}(L_k(x)) = \frac{n(L_k(x))}{\sum_i n(L_i(x))}. \tag{5}$$

However, the location prior $\mathbb{P}_L(x)$ is still unknown and the coverage of the gazetteer used by Yahoo Flickr is not large enough. In order to address these problems, we proposed an algorithm to inference location priors from web pages.

Intuitively, if the ancestral location senses or the offspring locations of the location sense $L_j(x)$ appear in document d_t, then the confidence of $c(L_j(x)|d_t)$ will be high. For example, in Fig. 1, there are 1 hidden node for the location sense `Pearl Harbor, Honolulu, HI`, and 2 hidden nodes for the location sense `Pearl harbor, Juneau, AL`. Therefore, `Pearl Harbor, Honolulu, HI` is more likely to be the correct location sense.

Based on this observation, the confidence of $c(L_j(w_i)|d_t)$ can be formulated as

$$c(x^k|d_t) = \frac{1 + \sum_{w_p^q} \omega^{d(x^k, w_p^q)} + \sum_{w_m^n} \omega^{d(x^k, w_m^n)}}{1 + \sum_{w_u^v} \omega^{d(x^k, w_u^v)} + \sum_{w_s^t} \omega^{d(x^k, w_t^t)}}, \tag{6}$$

where $x^k = L_k(x)$ is the kth location sense of geographic name x, $w_p^q = L_q(w_p)$ is the appeared ancestral location sense of x^k, $w_m^n = L_n(w_m)$ is the appeared offspring location sense of x^k, $w_u^v = L_v(w_u)$ is the ancestral location sense of x^k, $w_s^t = L_t(w_s)$ is the offspring location sense of x^k, ω is the decay factor.

For example, in Fig. 1,

$$\begin{aligned}
c(\texttt{Hawaii}|d_t) &= \frac{1+\omega+2\omega^2}{1+\omega+2\omega^2} = 1 \\
c(\texttt{Hawaii, HI}|d_t) &= \frac{1+2\omega}{1+2\omega} = 1 \\
c(\texttt{PearlHarbor, AL}|d_t) &= \frac{1}{1+\omega+\omega^2} \\
c(\texttt{PearlHarbor, HI}|d_t) &= \frac{1+\omega^2}{1+\omega+\omega^2}
\end{aligned}$$

In the example, the node `USA` is not used since it is shared by all location senses. Once $c(x^k|d_t)$ is computed, $\mathbb{P}(x^k|d_t)$ and $\mathbb{P}_L(x|d_t)$ can be calculated by

$$\mathbb{P}(x^k|d_t) = \frac{S(x^k|d_t)}{\sum_i S(x^i|d_t)} \tag{7}$$

$$\mathbb{P}_L(x|d_t) = \sum_k \mathbb{P}(x^k|d_t)c(x^k|d_t) \tag{8}$$

$$\mathbb{P}_L(x) = \frac{\sum_t I(x|d_t)p_L(x|d_t)}{\sum_t I(x|d_t)} \tag{9}$$

where $I(x|d_t)$ indicates that whether x is mentioned by document d_t or not. We can see that the computation of $\mathbb{P}_L(x)$ relays on $\mathbb{P}(x^k)$, and the computation of $\mathbb{P}(x^k)$ also relays on $\mathbb{P}_L(x)$. So an iterative method is proposed to compute $\mathbb{P}_L(x)$ and $\mathbb{P}(x^k)$ iteratively. We set the initial value of $\mathbb{P}_L(x) = 0.1$, and $\mathbb{P}(x^k) = 1/|x^k|$, where $|x^k|$ is the count of all possible location senses for geographic name x.

2.3 Inference Context Term Prior from Search Results

A location usually may also associate with some representative keywords. For example, people who go to Hawaii may also mention some relevant words, like beach, surfing, diving, sea food, and etc. People who go to Alaska may also mention some relevant words, like polar bear, ski, glacier and etc. This information is invaluable for location disambiguation. For example, the word "Gary" is used to refer a person name. However, when words "steel" and "RailCats" are discovered in this nearby context, the word "Gary" is most likely the location name of `Gary, IN, US`.

However, discovering such context correlation knowledge is not easy. General approaches require a lot of labeled data and suffer from the coverage problem. In this paper, we observe that general Web search engine usually can do pretty good in finding relevant information. For example, we submit three queries, "Gary", "Gary, IN" and "Gary, MN". We could find that the search results from the first query are quite diverse, the results from the second query is mainly about the Gary City in Indiana state, and the results from the third query are mainly about the Gary City in Minnesota

Bi-gram	Top 10 patterns
Prefix	town of, arrive in, back to, here in, day in, stop in, live in, <punc> in, back in, park in
Postfix	<punc> we, to visit, <punc> and, border <punc>, is a, to see, for the, and the, for a, and then
Combofix	in <punc>, to <punc>, in and, to to, into <punc>, to and, from <punc>, toward <punc>, in for, from to

(<punc> means punctuations)

Table 1: Example Bi-gram patterns

state. Moreover, the count of search results also contains rich information.

In this paper, we use location senses and geographic names as queries, and download search result counts and snippets for future processing. Based on this data set, we convert the location context and query snippets to TF-IDF vectors and then calculate cosine similarity between the location context and query snippets. That is, let all possible occurring word be $w_1, \cdots, w_n$, then both the search result and the document will be regarded as an n-dimensional vector

$$D_i = \{d_{i1}, \cdots, d_{in}\} = \{\texttt{tf-idf}_{i1}, \cdots, \texttt{tf-idf}_{in}\}$$

and

$$\texttt{tf-idf}_{ij} = \frac{n_{ij}}{\sum_k n_{ik}} \cdot \log \frac{|D|}{\{D_n : w_i \in D_n\}}$$

Therefore,

$$\texttt{Similarity}_{ij} = \cos\langle D_i, D_j \rangle = \frac{\sum_k d_{ik} \cdot d_{jk}}{\sqrt{\sum_k d_{ik}^2} \cdot \sqrt{\sum_k d_{jk}^2}}$$

2.4 Inference from local linguistic context

Similarly to other statistical NLP efforts, we also use the local linguistic context to do location extraction. For example, a capitalized phrase leading with "the" or ends with the abbreviation of a state will be very strong evidence for a location. Negative patterns, such as "Mr." or "Mrs.", may also dhelp us tell human names from location names.

Among all linguistic features we used, Bi-gram feature is one of the most informative one. Bi-gram feature is used to measure the confidence of a detected name being a location given it's surrounding 2 words. For example,

$$\begin{aligned} &\mathbb{P}(X \text{ is a location} | \text{prefix} = \text{“arrive in”}) \\ = \quad &\frac{\#\{\text{“arrive in } \langle\text{location}\rangle\text{”}\}}{\#\{\text{“arrive in”}\}} \end{aligned}$$

Three kinds of bi-gram models: prefix, postfix and combofix are used in our method. Table 1 listed top 10 scored patterns we deduced for all 3 kinds of bi-grams mined from one million sentences with location ground truth.

3. ITERATIVE LOCATION EXTRACTION FRAMEWORK

Traditional location extraction problem can be formulated as following sub-problems, 1) location name detection by inferencing $P_L(w_i|d)$, 2) location entity disambiguation by inferencing $P(L_j(w_i)|d)$. If we considering the interdependency between location name detection and location entity disambiguation, these two sub-problems can be solved by inferencing $P_L(w_i|d, S(d))$ and $P(L_j(w_i)|d, G(d))$, where $S(d)$ is the list of all location entities contained in document d, and $G(d)$ is the list of all location names contained in document d.

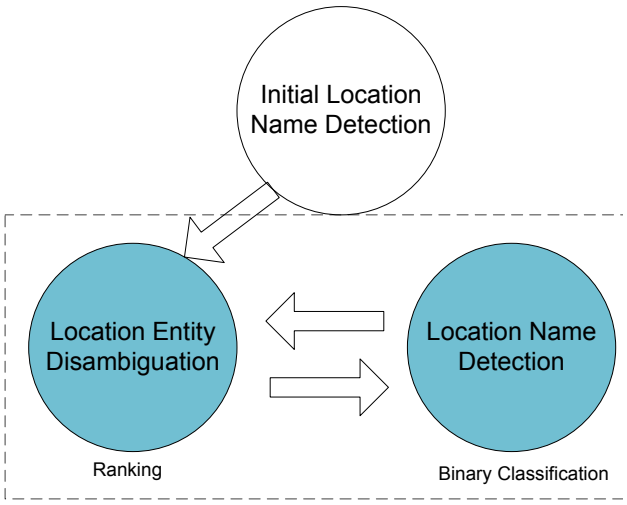

Figure 3: Location extraction is an iterative procedure

Obviously, in this framework, these two sub-problems are interdependent and become a chicken and egg dilemma.

In order to solve this problem, an iterative algorithm is proposed, as shown in Fig. 3,

1. Initial detection by inferencing $g_i = P_L(w_i|d)$
2. select a set of location terms $\hat{G}(d) = \{w_i | g_i > \alpha\}$, where α is a predefined threshold.
3. Inference the location sense $s_{i,j} = P(L_j(w_i)|d, \hat{G}(d))$
4. select a set of location entity $\hat{S}(d) = \{L^*(w_i) | w_i \in \hat{G}(d)\}$, where $L^*(w_i) = \text{argmax}_j s_{i,j}$
5. the location name detection by inferencing $g_i = P_L(w_i|d, \hat{S}(d))$
6. Goto step 2 until converges.

In this framework, the location evidence set Gd is first estimated by $\hat{G}(d)$ using initial detection, then based on these context information, we iteratively update the location evidence set $\hat{G}(d)$, until it is converged. The converge criteria we used here is the Jaccard similarity between the current location evidence set and the previous location evidence set. If the similarity score above a given threshold θ, the algorithm will be terminated. Usually, this algorithm quickly converged after 1 iteration running. In this paper, we use linear SVM to solve step 1) and 3), use linear rank SVM to solve step 2). Actually, this method can be extended to other location extraction algorithm which provides confidence output.

4. EXPERIMENTS

To demonstrate the performance of the proposed algorithm, we implement a location extractor called GeoScope.

Gazetteer	Chinese	English
#Name	475406	812757
#Location	630760	896598
Avg. #location per name	1.72	1.37
#Locations with alias	189002	129155
Avg. #alias	2.01	2.69

Table 2: Statistic of Gazetteers

	Traning	Evaluation
#Document	549	287
Avg. Length	11.6KB	11.7KB
	2484 Tokens	2497 Tokens
#Candidates are locations	18069	9760
#Candidates aren't locations	25629	12983
#Locations	3449	2175

Table 3: Training and Evaluation Document Set

In the classification step, we use the linear Support Vector Machine, and in the ranking step, we use the RankSVM [10]algorithm. we collect data from the Web to train our model. Total 500k blog posts have been crawled from Internet. To demonstrate the ability of multi-language location extraction, we build a location extractor support both English and Chinese languages. The gazetteers we used are from US Geological Survey [4] and AutoNavi [1].

We also hire 10 native labelers from a third-party company to label 2000 documents with geographic location ground truth for training and evaluation. Each document is labeled by at least 3 different individuals for cross validation. We also crawl over 92.3GB search result snippets from Bing search engine. Table 2 shows a simple statistic of our gazetteers.

Yahoo Placemaker and Metacarta GeoTagger, are two state-of-the-art commercial online geographical information retrieving services. Due to the low coverage issue of both Placemaker and GeoTagger in Chinese location, we conduct the experimental comparisons on English US locations set. The ground truth we used for these evaluations is our evaluation document set mentioned above.

For the instance of English version, we have total 836 documents labeled by reviewers. Table 3 is some basic information of this document set.

4.1 Performance Evaluations

In this section, we implement three location extractors on different feature sets, 1)Location extraction using only prior and linguistic features, 2) location extraction using prior, linguistic and location context features, and 3) location extraction using prior, linguistic, location context and term context features. The experimental results are shown in Fig. 4, from which we can observe that location context and term context features can substantially improve the accuracy of location extraction.

We also conduct an other set of experiments to compare the location extraction performance on different iteration stages. Table 4 shows that 1) the first round of iteration can significantly improve the performance of location extraction, and 2) the iterative location extraction method converges quickly in the second round of iteration.

4.2 Performance Comparisons with Placemaker

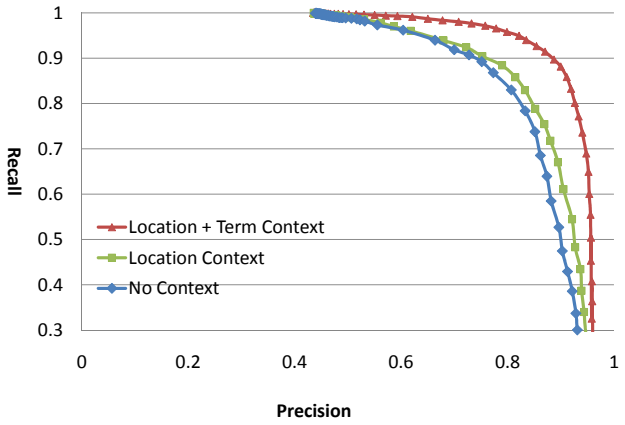

Figure 4: Iterative location extraction using contextual features

Step	Precision	Recall
1	72.77%	90.38%
2	88.40%	87.20%
3	89.61%	86.27%

Table 4: Performance in Iterating Classification

Yahoo Placemaker is a freely available geoparsing Web service offered by Yahoo. Provided with text, the service identifies places mentioned in text, disambiguates those places, and returns unique identifiers for each, as well as the occurrence of the places in the text[3]. It uses Yahoo GeoPlanet as it's gazetteer and identifier system. All details of the GeoPlanet gazetteer, including names, aliases and administrative hierarchy information, is also published freely online. Table 5 is a simple statistic of the GeoPlanet gazetteer.

Since the administrative hierarchy is available, we can align GeoPlanet gazetteer with our gazetteer. Finally we get 86308 locations and 157234 location names in the gazetteer intersection. Actually, among all 8081 hits labeled by Placemaker from our evaluation documents, only 3157 of them are locations which can be aligned to our gazetteer, and also among all 9760 locations in the ground truth, only 6243 of them are aligned with GeoPlanet gazetteer. We make performance comparisons on the recall rate and the precision rate. The results are shown in the right chart of Fig. 5.

In order to avoid the by-effect from gazetteer matching, we hire 20 labelers to provide blind reviews for the location extraction results from Yahoo Placemaker and GeoScope manually. Using this data, performance comparisons can be made directly on the location extraction results, which are shown in the left chart of Fig. 5.

From these results, we could observe that our algorithm is consistently better than Yahoo Placemaker in both experiments.

#Name	424620
#Location	404851
Avg. #location per name	1.46
#Location have alias	163345
Avg. #alias	2.30

Table 5: Statistic of Yahoo GeoPlanet gazetteer

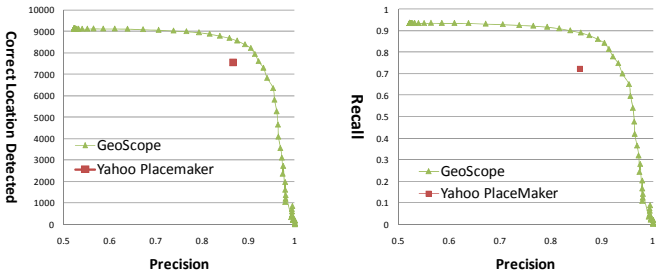

Figure 5: Performance comparisons with Yahoo Placemaker

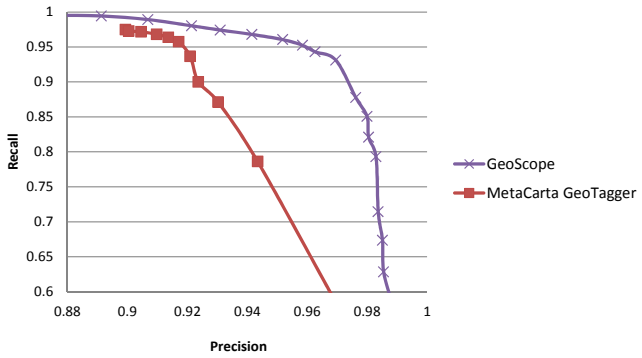

Figure 6: Performance comparisons with GeoTagger on geographic location detection

4.3 Performance Comparisons with GeoTagger

Metacarta is a famous company in the field of providing geographic solutions. It also has a geographical information retrieving product called GeoTagger [2] which can be accessed freely through Metacarta OnDemand web service. There are 763986 locations and 651659 names have been aligned between our gazetteer and GeoTagger's, covered most of the gazetteers. Therefore, we conduct our experiments on this gazetteer intersection. Fig. 6 and Fig. 7 show that the proposed algorithm significantly outperform the GeoTagger in both location detection and location entity disambiguation.

4.4 Performance on News Articles

Different from the blog posts, news articles are usually much shorter and contain fewer geographical references. Therefore, contextual information in these documents is less reliable. We randomly select 100 short news articles and build another document set. The average length of these articles is 2k bytes (427 words). Evaluation has been made between our method and Yahoo Placemaker. Results are all reviewed by third-party reviewers for both methods. The result shows in Table 6. From this table, we could observe that the performance of two methods are comparable.

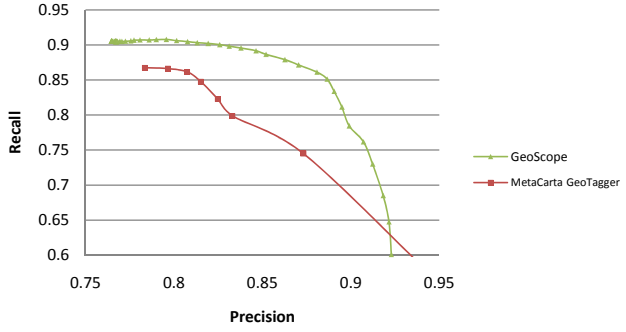

Figure 7: Performance comparisons with GeoTagger

	#Detected	#Correct	Precision
Yahoo Placemaker	559	337	60.28%
Our Method	507	311	61.34%

Table 6: Evaluation on News Articles

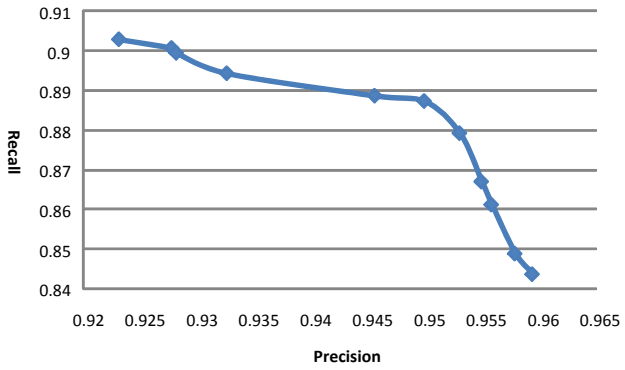

Figure 8: Performance on Chinese Documents

4.5 Performance of the Chinese Version

Our algorithm can be easily applied to non-English documents. In this section, we port our algorithm to Chinese location extraction. However, the Chinese gazetteers used in GeoTagger and Placemaker are very limited (2k- locations), which is not comparable with our approach (900k Chinese locations). Fig. 8 shows that our method achieves 90%+ recall rate and precision rate.

4.6 Discussion

Fig. 4 shows that the proposed contextual information mined from Web resources can substantially improve the location performance. In Fig. 5, Fig. 6 and Fig. 7, we also observe that the proposed algorithm provides better performance than the state-of-the-art industrial approaches. However, when applied to new articles, which are shorter and containing few contextual information, the performance improvement of proposed algorithm is minor. Fig. 8 shows that our algorithm can be easily extended to non-English location extraction.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel location extraction algorithm to utilize rich semantic features mined from Web. The experimental results show that semantic features encode human real-world knowledge and can effectively improve the performance of location extraction. Besides the Flickr GeoTags and search engine snippets, there are also some valuable source of semantic information, such as, local forum, Wikipedia articles, WordNet and etc. The study of using

this information will be our future work.

6. ACKNOWLEDGEMENT

I would like to thank all members of the GeoScope team for a creative and stimulating project environment. In particular, I am grateful to Xing-Rong Cheng, Jiang-Ming Yang, and Xiao-Long Ma for many fruitful discussions, creative ideas, and for being reliable project partners.

7. REFERENCES

[1] AutoNav, http://www.autonavi.com/en.

[2] GeoTagger, http://www.metacarta.com/products-platform-geotag.htm.

[3] PlaceMaker, http://developer.yahoo.com/geo/placemaker/.

[4] USGS, http://geonames.usgs.gov/.

[5] E. Amitay, N. HaráEl, R. Sivan, and A. Soffer. Web-a-where: Geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280, 2004.

[6] T. J. Brunner and R. S. Purves. Spatial autocorrelation and toponym ambiguity. In *Proceeding of the 2nd international workshop on Geographic information retrieval*, pages 25–26, 2008.

[7] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, pages 9–16, 2006.

[8] J. Ding, L. Gravano, N. Shivakumar, and G. Inc. Computing geographical scopes of web resources. In *Proceedings of the 26th International Conference on Very Large Databases, VLDB 2000*, pages 545–556, 2000.

[9] J. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, 2005.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 142, 2002.

[11] J. Leveling., S. Hartrumpf., and D. Veiel. University of Hagen at GeoCLEF 2005: Using semantic networks for interpreting geographical queries. In *Working Notes for the GeoCLEF 2005 Workshop*, 2005.

[12] H. Li, R. K. Srihari, C. Niu, and W. Li. Location normalization for information extraction. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, 2002.

[13] H. Li, R. K. Srihari, C. Niu, and W. Li. Infoxtract location normalization: a hybrid approach to geographic references in information extraction. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 39–44, 2003.

[14] M. Nissim, C. Matheson, and J. Reid. Recognising geographical entities in Scottish historical documents. In *Proceedings of the Workshop on Geographic Information Retrieval at SIGIR 2004*, 2004.

[15] S. Overell. *Geographic Information Retrieval: Classification, Disambiguation and Modelling*. PhD thesis, Imperial College London, 2009.

[16] S. Overell and S. Ruger. Using co-occurrence models for placename disambiguation. *International Journal of Geographical Information Science*, 22(3):265–287, 2008.

[17] T. Poibeau and L. Kosseim. Name extraction from non-journalistic texts. In *Computational Linguistics in the Netherlands*, pages 144–157, 2001.

[18] T. Qin, T. Liu, X. Zhang, D. Wang, and H. Li. Global ranking using continuous conditional random fields. In *Proceedings of NIPS*, volume 8, 2008.

[19] E. Rauch, M. Bukatin, and K. Baker. A confidence-based framework for disambiguating geographic terms. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 50–54, 2003.

[20] D. A. Smith and G. S. Mann. Bootstrapping toponym classifiers. In *HLT-NAACL Workshop on Analysis of Geographic References*, pages 45–49, 2003.

[21] R. Volz, J. Kleb, and W. Mueller. Towards ontology-based disambiguation of geographical identifiers. In *Proceedings of the 16th international conference on World Wide Web*, 2007.

[22] C. Wang, X. Xie, L. Wang, Y. Lu, and W. ying Ma. Detecting geographic locations from web resources. In *Proceeding of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005.

[23] L. Wang, C. Wang, X. Xie, J. Forman, Y. Lu, W.-Y. Ma, and Y. Li. Detecting dominant locations from search queries. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 424–431, 2005.