Extracting Focused Locations for Web Pages

Qingqing Zhang, Peiquan Jin, Sheng Lin, and Lihua Yue

University of Science and Technology of China
jpq@ustc.edu.cn

Abstract. Most Web pages contain location information, which can be used to improve the effectiveness of search engines. In this paper, we concentrate on the focused locations, which refer to the most appropriate locations associated with Web pages. Current algorithms suffer from the ambiguities among locations, as many different locations share the same name (known as GEO/GEO ambiguity), and some locations have the same name with non-geographical entities such as person names (known as GEO/NON-GEO ambiguity). In this paper, we first propose a new algorithm named *GeoRank,* which employs a similar idea with *PageRank* to resolve the GEO/GEO ambiguity. We also introduce some heuristic rules to eliminate the GEO/NON-GEO ambiguity. After that, an algorithm with dynamic parameters to determine the focused locations is presented. We conduct experiments on two real datasets to evaluate the performance of our approach. The experimental results show that our algorithm outperforms the state-of-the-art methods in both disambiguation and focused locations determination.

Keywords: Web Search, Geographical information, GEO/GEO ambiguity, GEO/NON-GEO ambiguity, Focused locations.

1 Introduction

Web search engines such as Google and Bing have been an important part in people's life. However, existing search engines do not pay enough attention to the location information in Web pages. For example, it is difficult to express queries like "to find the retailer promotion about Nike in Beijing" in Google. On the other side, location, or in other words, the spatial dimension, is one of essential characteristics of information, and most Web pages are associated with certain locations, e.g., news report, retailer promotion and so on. A recent study in the literature [21] reported that among 2,500 queries, 18.6% of them contained a geographic predicates and 14.8% of them included a place name. Therefore, how to extract locations for Web pages and then use them in Web search process has been a hot and critical issue in current Web search.

As a Web page usually contains two or more location words, it is necessary to find the focused locations of the Web page. The focused locations represent the most appropriate locations associated with contents of a Web page. Generally, we assume that each Web page has several focused locations. The most difficult issue in determine focused locations is that there are GEO/GEO and GEO/NON-GEO

L. Wang et al. (Eds.): WAIM 2011 Workshops, LNCS 7142, pp. 76–89, 2012.

ambiguities existing in Web pages. The GEO/GEO ambiguity refers that many locations can share a single place name. For example, Washington can be 41 cities and communities in the United States and 11 locations outside [5]. The GEO/NON-GEO ambiguity refers that a location name can be used as other types of names, such as person names. For example, Washington can be regarded as a person name as George Washington and as a location name as Washington, D.C. Mark Sanderson's work [22] shows that 20%-30% extent of error rate in location names disambiguation was enough to worsen the performance of the information retrieval methods. Due to those ambiguities in Web pages, previous research failed to reach a satisfied performance in focused locations extraction.

On the other side, it is hard to resolve the GEO/GEO and GEO/NON-GEO ambiguities as well as to determine the focused locations of Web pages through the widely-studied named entity recognition (NER) approaches. Current NER tools in Web area aim at annotating named entities including place names from Web pages. However, although some of the GEO/NON-GEO ambiguities can be removed by NER tools, the GEO/GEO disambiguation is still a problem. Furthermore, NER tools have no consideration on the extraction of the focused locations of Web pages. Basically, the NER tools are able to extract place names from Web pages, which can be further processed to resolve the GEO/GEO ambiguities as well as the GEO/NON-GEO ones. Thus, in this paper we will not concentrate on the NER approaches but on the following disambiguation and focused locations determination. Those works differ a lot from traditional NER approaches.

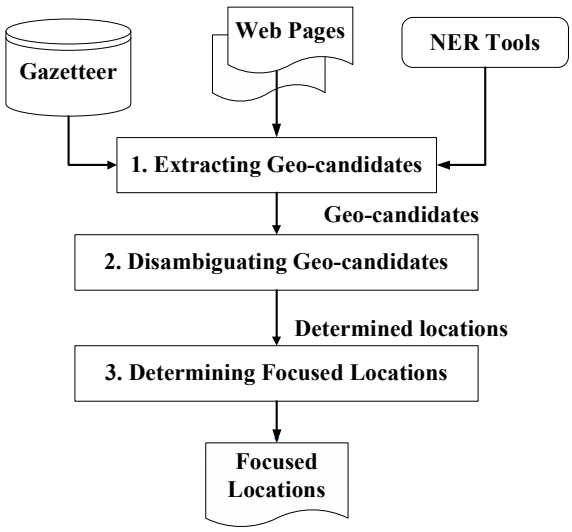


Fig. 1. The general process to extract focused locations from Web pages

Figure 1 shows the general process to extract focused locations from Web pages, in which we first extract geo-candidates based on Gazetteer and NER (named entity recognition) techniques. After this procedure, we get a set of geo-candidates. In this set, the relative order of candidates is the same as that in the text. Here, geo-candidates are just possible place names, e.g., “Washington”. Then, we run the

disambiguation procedure to assign a location for each GEO/GEO ambiguous geo-candidate and remove GEO/NON-GEO ambiguous geo-candidates. Location means a concrete geographical place in the world, e.g.: USA/ Washington, D.C. As a geo-candidate may refer to many locations in the world, the GEO/GEO disambiguation will decide which is the exact location that the geo-candidate refers to and, the GEO/NON-GEO disambiguation is going to determine whether it is a location or not. Finally, we present an effective algorithm to determine focused locations among the resolved locations.

The main contributions of the paper can be summarized as follows:

(1) We propose the *GeoRank* algorithm to resolve the GEO/GEO ambiguity and a heuristic approach to remove the GEO/NON-GEO ambiguity (Section 3). Particularly, the *GeoRank* algorithm uses a similar way as *PageRank* but focused on the determination of the exact location associated with a specific geo-candidate. And the experimental results demonstrate that *GeoRank* outperforms previous methods.

(2) We present an effective algorithm to determine focused locations for Web pages (Section 4), which uses dynamic parameters when computing other locations' contribution to a given location. Compared with the state-of-the-art algorithms with static parameters, our algorithm is more reasonable in computing the importance of locations and has better performance.

(3) We carry out experiments based on real datasets to evaluate the performance of our disambiguation algorithm as well as the algorithm to determine focused locations .

2 Related Work

Disambiguation is usually implemented by using some information in the text such as zip code, phone number and so on. Volz et al. [24] proposed a two-step method, which first used context information to narrow candidates and then ranked the left candidates primarily based on weights according to concepts. Rauch et al. [28] proposed a confidence-based approach. Silva et al. [18] used some classification rules and ontology-based classification such as feature type to disambiguate and with the help of relationships among different geographical concepts, then they used a variation of *PageRank* algorithm to get the focused locations. Place name patterns were studied in SASEIC [12], in which they first examine possible patterns in the Web page, and with the help of these patterns and hierarchical structure of places they get focus of the page. Ding et al. [13] used hyper-links to help decide the page focus. Markowetz et al. [17] and Sobhana et al. [23] made use of the best one of the biggest town first methods and co-occurrence models to remove geographical ambiguity. Andogah et al. [3] proposed a totally different way, with the help of geo-candidate frequency, place type and other features In MyMoSe [25], a K-partite graph for disambiguation was proposed, which used a score-based approach to determine focused locations. There are also other works that employed heuristics in disambiguation [15, 16].

There are also a lot of related works in locations detection [10, 20, 26, 27]. Web-a-where is a four-step heuristics algorithm to determine focused locations for Web

pages [10], in which all names were assigned a location with a confidence score. Based on those confidence scores, as well as other information such as frequency, location relationships and so on, the focused locations of a Web page are extracted. However, Web-a-where adopts fixed parameters and thresholds, which are not suitable for different kinds of Web pages. The evidence-based method is an effective algorithm for geo-candidates disambiguation [26], which makes use of metric relation, topological relation and typological relation between an ambiguous geo-candidate and other co-occurring geo-candidates in the context. Those co-occurring candidates are regarded as the evidences of a geo-candidate, which are fused by the Dempster-Shafer (D-S) theory. However, both of [10] and [26] did not consider the changing confidence that a geo-candidate impacts on other ones, which will lead to bad performance of disambiguation. As shown in our experimental results, the evidence-based method has a comparable performance with Web-a-where in resolving place names ambiguity.

3 Geo-Candidates Disambiguation

3.1 The *GeoRank* Algorithm for Resolving the GEO/GEO Ambiguity

3.1.1 Basic Idea

As Fig.1 shows, we have a set of geo-candidates at present before the disambiguation procedure. We first assume that all geo-candidates are associated with the locations in the Web page. Basically, we assume there are *n* geo-candidates and totally *N* locations that *n* geo-candidates can have in a Web page, the GEO/GEO disambiguation problem can be formalized as follows: *Given a specific geo-candidate G, determining the most appropriate location among its possible locations.*

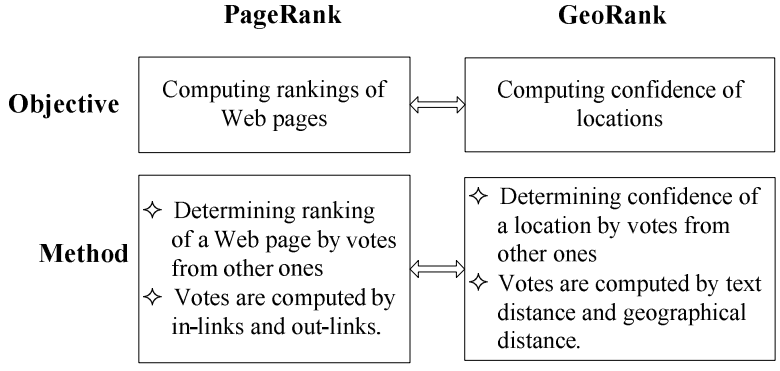


Fig. 2. PageRank vs. GeoRank

We use a general idea similar to *PageRank* to resolve the GEO/GEO ambiguity, which is named *GeoRank*. The *PageRank* algorithm introduced an iterated voting process to determine the ranking of a Web page. We also regard the GEO/GEO disambiguation in a Web page as a voting process. Figure 2 shows the similar problem definition between *PageRank* and our *GeoRank* algorithm. Specially, in

GeoRank, nodes are the locations corresponding to geo-candidates and the linkages are the evidence contributed by the locations each other. The higher score one location gets, the higher confidence it is the right location that the geo-candidate refers to.

In detail, as a geo-candidate can give more evidence to the one near to it in a Web page (text distance) and a location can give more evidence to the one near to it in the geographic context (geographical distance), we first construct a matrix involving all locations, whose values are scores of each location of each geo-candidate voted by other ones that belong to different geo-candidates.

3.1.2 Vote Computation

For simplification, all the possible locations associated with a certain geo-candidate have totally one vote that can be endowed to locations of other geo-candidates. A vote does not mean 1 in actual, it has something to do with the total number of geo-candidates, which we will discuss in section 3.1.3. Initially, all the locations of a geo-candidate have the same percentage of vote, namely 1/ *size* (*L*), L is the set of locations corresponding to this geo-candidate. For example, if a geo-candidate G has 10 possible locations; *size* (*L*) is 10, so their initial percentage of vote is 1/10.

Our *GeoRank* algorithm uses text distance and geographical distance to compute the percentage of vote that is contributed by locations of other geo-candidates to a certain location of a geo-candidate. In particular, the text distance is defined between two geo-candidates, while the geographical distance is defined between two locations of two different geo-candidates. The smaller those distances are, the more evidence they will give to each other.

Differing from traditional text distance that refers to the count of characters between two words, we use the term *relative text distance* to define the text distance between geo-candidates.

Definition 3.1: Relative Text Distance (RTD). Given two geo-candidates G_i *and* G_j in a Web page, their relative text distance is defined as *RTD* (G_i, G_j), which is determined by the count of sentences in the Web page between G_i *and* G_j.

Moreover, if two geo-candidates appear in the same sentence, the *RTD* value will be set to their distance in the geo-candidates list. It's reasonable that two geo-candidates appear in one sentence should have stronger evidence to each other. May be many geo-candidates appear more than once, we define *RTD* (G_i, G_j) as the smallest one among them.

Based on the definition of *RTD*, we can define the vote percentage of a geo-candidate to other geo-candidates.

Definition 3.2: Geo-candidate Vote. Given a geo-candidate G_i, and a set of ambiguous geo-candidates (G_1, G_2… G_m), G_i's vote to G_j is defined as *GV* (G_i, G_j):

$$GV(G_i, G_j) = \left(\frac{1}{RTD(G_i, G_j)}\right) / \sum_{k=1,k\neq i}^{m} \frac{1}{RTD(G_i, G_k)} \quad \blacksquare$$

This formula means that a geo-candidate have fixed percentage of vote, which it gives to other geo-candidates according to their RTD value. We make a normalization to assure that all of G_i's vote is spread to others.

For example, if there are three ambiguous geo-candidates G_1, G_2 and G_3 in a Web page, with $RTD(G_1, G_2)=1$ and $RTD(G_1, G_3)=2$, we can get that $GV(G_1, G_2)=2/3$ and $GV(G_1, G_3)=1/3$, which means G_2 will get 2/3 vote from G_1 while G_3 will get 1/3 vote. The definition of geo-candidate vote implies that a certain geo-candidate has more impact on its neighbors in the text, i.e., those geo-candidates with small *RTD* values.

As we want to finally compute the location-to-location vote, we need to divide the geo-candidate vote among the locations associated with the geo-candidate. So we introduce the geographical distance based method to deal with this issue. Since each geo-candidate appears in the gazetteer, which can be formally represented as a taxonomy tree, we can represent a location in the taxonomy tree as a structural sequence. For example, in the example in Fig.3, the location "*Peak Stone*" can be represented as "*USA/Massachuse/Peak Stone*". We then have the following observation that if two locations in the taxonomy tree have the same left prefix they tend to be located very closely. Therefore, we define the inverse geographical distance between two locations as follows.

Definition 3.3: Inverse Geographical Distance. Given two locations, *loc* of geo-candidate G_i and *loc'* of geo-candidate G_j, the geographical distance between *loc* and *loc'* is defined as *GD (loc, loc'),* and inverse geographical distance is defined as *IGD (loc, loc')*, which refers to the maximal count of the same left prefix between *loc* and *loc'*. A larger IGD value means two locations are nearer. ■

Definition 3.4: Location Vote. Given a location *loc* of geo-candidate G_i and an ambiguous geo-candidate G_j, L_j is the set of locations G_j corresponding to, and $loc' \in L_j$, the vote of *loc* to *loc'* is defined as *LV(loc, loc')*:

$$LV(loc, loc') = GV(G_i, G_j) * (IGD(loc, loc') / \sum_{loc'' \in L_j} IGD(loc, loc''))$$ ■

The location vote in Definition 3.4 indicates that the vote of *loc* to G_j is divided among all the possible locations associated with G_j according to their inverse geographical distances from *loc*. We use $GV(G_i, G_j)$ instead of each G_i's location vote, because each location's vote will be reflected when the confidence vector multiply the matrix, which we will discuss in the next section. This formula indicates that locations with larger the IGD value will get more percentage of vote from *loc*. In case that all the inverse geographical distances equal zero, we divide the $GV(G_i, G_j)$ among all the locations of G_j uniformly, i.e., $LV(loc, loc' \in L_j) = GV(G_i, G_j) / size(L_j)$. It indicates that *loc* cannot give any evidence to G_j, and we record this, which will be used in GEO/NON-GEO disambiguation.

3.1.3 The GeoRank Algorithm

GeoRank mainly consists of three stages (as shown in Fig.3):

(1) On the first stage (line 1 to 4), it computes the geo-candidate vote as well as the location vote, based on the relative text distance and inverse geographical distance which are defined in Section 3.1.2.

Algorithm *GeoRank*

Input: the set of geo-candidates $G = \{G_1, G_2 \ldots G_n\}$, the set of location sets $L = \{L_1, L_2, \ldots, L_n\}$, where L_i is the set of all the possible locations associated with G_i.

Output: the set of locations $D = \{D_1, D_2 \ldots D_n\}$, where D_i is the disambiguated location associated with G_i.

Preliminary: n is the count of geo-candidates, and N is the count of all possible locations associated with n geo-candidates.

/ Computing geo-candidate vote and location vote */*

1: **for** each $G_i \in G$ & $G_j \in \{G - G_i\}$ & G_j is ambiguous **{**
2: compute $RTD(G_i, G_j)$ and then $GV(G_i, G_j)$;**}**
3: **for** each $G_i \in G$ & $G_j \in \{G - G_i\}$ & G_j is ambiguous & $loc \in L_i$ & $loc' \in L_j$ **{**
4: compute $GD(loc, loc')$ and then $LV(loc, loc')$;**}**

/ Initializing the matrix for all the locations and the confidence vector */*

5: Initializing an $N \times N$ matrix M, with each location occupies one row and one column. The initial state of M is set by the following rule. {
6: **for** each location *loc* and *loc'* {
7: **if** *loc* = l*oc'* **then** $M[i, j] \leftarrow 0$
8: **else** $M[i, j] \leftarrow LV(loc, loc')$;**}}**
9: $M = (1-\alpha)M + \alpha S$; *// modify M according to Bryan et al.* [29]
10: Constructing the confidence vector $V = (v_1, v_2, \ldots, v_N)$, For each location loc_i, $v_i = 1/(n*count(G_k))$, where G_k is the geo-candidate loc_i is associated with, and $count(G_k)$ refers to the count of locations associated with G_k.

/ Determining the exact location of geo-candidate */*

11: **while** V does not converge **{**
12: $V = M * V$;
13: normalizing V so that $\sum_1^N v_i = 1$; **}**
14: Normalizing all the locations' vector values of each geo-candidate to make their sum to be 1.
15: **for** each $G_i \in G$ **{**
16: **for** each location $loc \in L_i$ **{**
17: **if** the *loc*'s vector value in $V > \delta$ **then** {
//i.e., δ *is* a predefined threshold
18: $D_i \leftarrow loc$; **exit for**; **}**
19: $D_i \leftarrow$ *use the server location and default meaning to help decide*;**}}**
20: **return** D;
End *GeoRank*

Fig. 3. The *GeoRank* Algorithm

(2) On the second stage (line 5 to 10), it initializes the matrix M for all the locations associated with each geo-candidate, as well as the initial confidence vector V. Generally, the vector represents each location's confidence of a geo-candidate. At first, we assume that each location of a geo-candidate has the same confidence. To make it adaptive to *PageRank*, the sum of all elements in V will be 1;

(3) Then on the third stage (line 11 to 20), we update the vector literately by introducing the influence of M into the confidence vector. The iteration process is similar to *PageRank*. According to Bryan et al. [29], the vector V will converge after several iterations, as we modify the matrix as $M = (1-\alpha)M + \alpha S$, S denotes an $N*N$ matrix with all entries $1/N$, M is column-stochastic and irreducible, according to *Perron–Frobenius theorem*, the vector V will finally converge and reach a stable state, which is not influenced by the initial values of the vector. In the experiment we set α as 0.1.

In the algorithm, we use a threshold δ, which is 0.6 in the implementation, to determine whether a location is the most relevant one for the given geo-candidate. A 0.6 threshold means the location has a confidence of 60% to be the location that the geo-candidate indicates. In case that all the locations associated with the given geo-candidate have vector values (confidence) less than the threshold, which implies that no location of the geo-candidate can be determined in the Web page, then we use the server location of the Web page as a filter and then the default sense to determine the real meaning of geo-candidates. We use the one that has the largest population as its default sense.

3.2 The Heuristic Algorithm for Resolving GEO/NON-GEO Ambiguity

Named entity recognition tools usually can remove some types of the GEO/NON-GEO ambiguities in a Web page. In order to get an improved performance, we propose two additional heuristics in the paper to further resolve GEO/NON-GEO ambiguities. Note these rules are based on the *GeoRank* algorithm we discussed in Section 3.1.

Rule 1: When constructing the matrix M (see Fig.3), if locations of a geo-candidate gets score averagely from all locations of other geo-candidates, it is considered not a location. It is reasonable that none of any possible location of any other geo-candidate can give evidence to locations of this geo-candidate; it is possibly not a location.

Rule 2: After removing the GEO/GEO ambiguity, if a non-country location does not have the same country with any other location; it is considered not a location. Here we get the rule from our observation that a Web page is unlikely to mention a non-country location that does not share a same country with any other locations.

4 Determining Focused Locations

In this stage, we calculate the scores of all the locations after disambiguation, and then return the focused ones for the Web page. We consider three aspects when computing the scores of a location, namely the term frequency, position and the contributions from locations geographically contained by the location. An example of the latter aspect is that if there are many states of USA in a Web page, the location USA will receive contributions from those states, as those states are all geographically contained in USA and mentioning states explicitly means mentioning USA implicitly.

As a result, we use an explicit score to represent the term frequency of a location name, and an implicit score for the geographical containment. The score of a location is its explicit score plus its implicit score.

For location D_i, its explicit score, denoted as ES (D_i), is defined as the term frequency of D_i in the Web page.

Then we use the following heuristics to modify ES (D_i):

(1) If D_i follows on the heels of the other location D_j and D_i has some relationship with D_j, suppose D_j is the son or grandson of D_i, then we think the appearance of D_i in the page aims at emphasizing or explaining D_j, so we take 0.5 away from D_i and add it to D_j, i.e., $ES(D_i) = ES(D_i) - 0.5$, $ES(D_j) = ES(D_j) + 0.5$.

(2) If D_i appears in the title of a Web page, then we add half of SUM to D_i to emphasize this appearance, where SUM is the sum of all the ES values, as defined in the formula 4.1.

$$SUM = \sum_{i=1}^{n} ES(D_i) \tag{4.1}$$

For the implicit scores, since many locations appear in one Web page usually have some geographical relationships, we take this feature when computing the implicit score of a location. In particular, we add some contributions from those locations contained by the given location into the score. Suppose a location D_i contains n sub-locations in the gazetteer: S_1, S_2,…, S_n, and the former m sub-locations appear along with D_i in the Web page, then those m sub-locations will contribute to D_i. The implicit score of D_i is defined in the formula 4.2 and 4.3.

$$IS(D_i) = \sum_{k=1}^{m} (ES(S_k) + IS(S_k)) * \frac{m}{n * diff} \tag{4.2}$$

$$diff = \frac{avg(S_1, S_2, ..., S_m)}{\max(S_1, S_2, ..., S_m)} \tag{4.3}$$

Here, *diff* refers to the score difference among S_1, S_2,…, S_m, The average value of S_1, S_2,…, S_m must be less than or equal to the maximum value of them, so *diff* <= 1. If D_i contains no sub-locations, then $IS(D_i) = 0$.

Based on a Gazetteer, we can build a hierarchy geographical tree for locations. Then we start from the leaf nodes and compute the scores of all locations. Then we sort all locations according to their scores and partition locations into three groups by using a native clustering approach. The first group with highest scores is determined as the focused locations.

The difference between our algorithm and Web-a-where in [10] is that they employ a fix parameter when measuring the implicit score of a location, namely 0.7, while in our algorithm we use a dynamic parameter as *m*/ (*n***diff*). *m/n* means the more sub-locations of a location appear, the more possibly it will be a focused location and *diff* means the less difference of sub-locations' score, the more possibly it will be a focused location, this means that this Web page does not emphasis any sub-locations. Thus our algorithm is adaptive to the occurrence of locations that are geographically

related with the given location. Our experimental results demonstrate that our method has benefits by using the dynamic parameter.

5 Experiments

5.1 Datasets

We conduct experiments on real datasets to measure the performance of our algorithm in geo-candidate disambiguation and focused locations determination. Two real datasets are used in the experiments, an nj.gov dataset downloaded from http://www.nj.gov/ and a BBC dataset downloaded from http://www.bbc.co.uk/. For the geo-candidate disambiguation experiment, we choose Web-a-where [10] and the evidenced-based method [26] as the competitors of our *GeoRank* algorithm. For focused locations determination, we compare the performance between our approach and Web-a-where [10]. As surveyed in [14], Web-a-where [10] has the best performance in focused location extraction for Web pages compared with other competitor methods. Therefore, it is meaningful to conduct comparison experiment with Web-a-where.

5.2 Pre-processing

5.2.1 Gazetteer Construction

We first construct a gazetteer based on World Gazetteer [8]. Our gazetteer contains 320,707 place names and 56,665 alternate names. We store the following information about a location in Microsoft SQL Server 2008 database: *id, name, population, latitude, longitude* and *upper* (Here *upper* means its parent which is also represented as a taxonomy node).

5.2.2 Geo-Candidates Extraction

For geo-candidates extraction, we employ CCG (Cognitive Computation Group) [1] as the NER tool. After name entity tagging, we get a set of geo-candidates. Then we scan the set and check each element if it or its relatives appears in the gazetteer. The detailed process is as follows (suppose G_1 is a geo-candidate):

(1) Check G_1 if it appears in the gazetteer, if not found, go to (2);

(2) Remove phrase like "City of" or "City" and repeat the checking in the gazetteer. If G_1 is not found in the gazetteer, we delete it from the list.

5.3 Geo-Candidates Disambiguation

In this procedure, we run our algorithm, Web-a-where [10], and the evidence-based method [26] to resolve the ambiguity of geo-candidates. We first remove the unambiguous ones, i.e., those with only one entry in the gazetteer. Then we get 1990 ambiguous geo-candidates for the nj.gov dataset and 2488 for the BBC dataset.

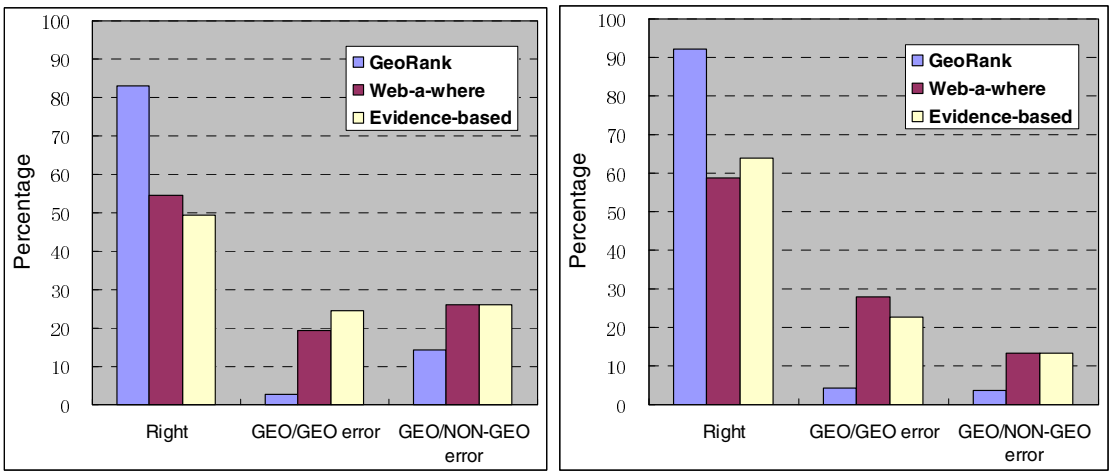
All the ambiguous geo-candidates are resolved by the three algorithms and the outputs are classified into three categories:

(1) *Right*: a geo-candidate is recognized rightly, it is assigned to a right location or it is not a location.

(2) *GEO/GEO error*: a geo-candidate with GEO/GEO ambiguity is not correctly resolved.

(3) *GEO/NON-GEO error*: a geo-candidate with GEO/NON-GEO ambiguity is not correctly resolved.

Figure 4 shows the percentages of the three categories of results for each algorithm (for simplification, *GeoRank* stands for both GEO/GEO and GEO/NON-GEO disambiguation), from which our *GeoRank* algorithm always has the best performance under two datasets and three metrics. In particular, GeoRank has a very low rate for the GEO/GEO error and GEO/NON-GEO error. This is because that *GeoRank* integrates into the disambiguation the confidence as well as its changing of all the locations for a geo-candidate. Another reason is due to its consideration on the text distance among all the geo-candidates appearing in a Web page. Furthermore, the heuristic rules used to reduce the GEO/NON-GEO ambiguity also contribute on the good performance.



(a) nj.gov (b) BBC

Fig. 4. Disambiguation results of *GeoRank*, Web-a-where and the evidenced-based method

5.4 Experiments on Determining Focused Locations

The results of focused locations determination are shown in Fig.5. As Fig.5 shows, we classify the results into four categories, namely *right*, *contain error*, *more or less error*, and *names error*. The definitions on those four metrics are as follows:

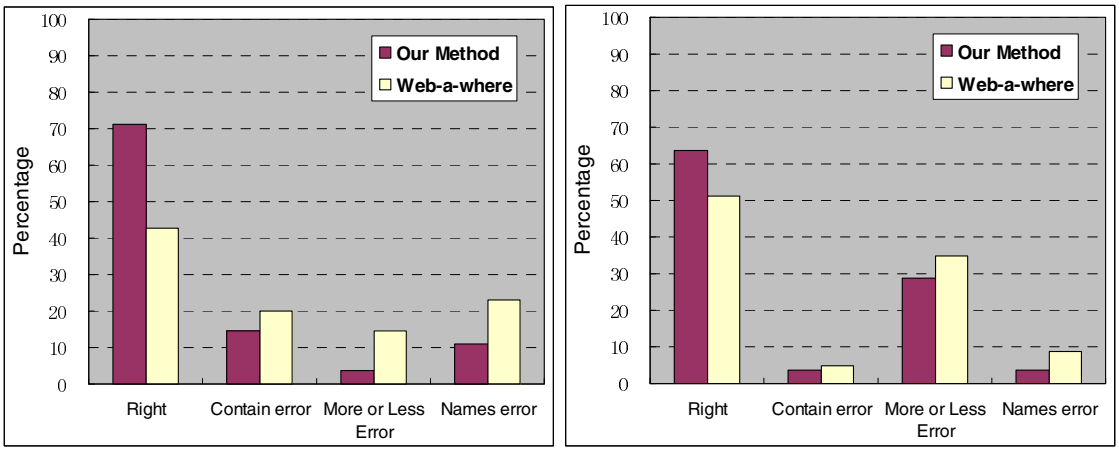
(1) *Right*: the focused locations are determined rightly.

(2) *Contain error*: the determined focused location has a larger or smaller geographical scope than the right one.

(3) *More or less error*: the number of focused locations is more or less than that of right ones.

(4) *Names error*: A wrong focused location is determined. This is mainly because of the former disambiguation error.

Here we only compare our algorithm with Web-a-where [10], as the evidence-based method does not have a procedure for focused locations determination. Figure 6 shows that our algorithm has not only better right rate but also lower error rate for all the three types of errors. According to our experimental results, "names error" is the most frequent error for Web-a-where [10], because of the error in disambiguation phrase. Web-a-where [10] also has a large number of "More or less errors", which are caused by their fixed parameter and thresholds. Differing from Web-a-where [10], we use dynamic parameter in our algorithm, which is demonstrated as a feasible approach to improving the performance of Web-a-where [10]. Another reason for the good performance of our algorithm is that we consider the positions of geo-candidates appearing in text into the computation of location scores.



(a) nj.gov (b) BBC

Fig. 5. Results of focused locations determination

6 Conclusions

In this paper, we concentrated on extracting focused locations from Web pages. In particular, we studied two issues, namely geo-candidates disambiguation and focused location extraction. We presented a new algorithm named *GeoRank* to resolve the GEO/GEO ambiguity and a framework to extract focused locations from Web pages. Experiments on different real datasets show that our approach has better performance than the state-of-the-art algorithms. We plan to make more comparisons by adjusting the parameter in our method and other approaches.

Acknowledgements. This work is supported by the National Science Foundation of China (no. 70803001), the Open Projects Program of National Laboratory of Pattern Recognition (20090029), the Key Laboratory of Advanced Information Science and Network Technology of Beijing (xdxx1005), and the USTC Youth Innovation Foundation.

References

1. Cognitive computation group, http://cogcomp.cs.illinois.edu/page/software (accessed in April 2011)
2. Gate, http://gate.ac.uk/ (accessed in April 2011)
3. Andogah, G., Bouma, G., Nerbonne, J., Koster, E.: Place name Ambiguity Resolution. In: Proc. of LREC, Marrakech Morocco, pp. 4–10 (2008)
4. Geonames, http://www.geonames.org (accessed in April 2011)
5. Washington, http://en.wikipedia.org/wiki/washington (accessed in April 2011)
6. United Nations department of economic and social affairs, http://unstats.un.org/unsd (accessed in April 2011)
7. Usgs geographic names information system (gnis), http://geonames.usgs.gov (accessed in April 2011)
8. World Gazetteer, http://www.world-gazetteer.com (accessed in April 2011)
9. Lingpipe, http://alias-i.com/lingpipe/ (accessed in April 2011)
10. Amitay, E., Har'El, N., Sivan, R., Soffer, A.: Web-a-where: geotagging Web content. In: Proc. of SIGIR, Sheffield, United Kingdom, pp. 273–280 (2004)
11. Anastacio, I., Martins, B., Calado, P.: A comparison of different approaches for assigning geographic scopes to documents. In: Proc. of the INForum 2009 (2009)
12. Chen, M., Lin, X., Zhang, Y., Wang, X., Yu, H.: Assigning geographical focus to documents. In: Proc. of Geoinformatics, Beijing, China, pp. 1–6 (2010)
13. Ding, J., Gravano, L., Shivakumar, N.: Computing geographical scopes of Web resources. In: Proc. of VLDB, Cairo, Egypt, pp. 545–556 (2000)
14. Gyle, A., Plaunt, C.: Gipsy: Automated geographic indexing of text documents. Journal of the American Society of Information Science 45(9), 645–655 (1994)
15. Leidner, J.L.: Toponym resolution in text: Annotation, evaluation and applications of spatial grounding of place names. PhD dissertation, University of Edinburgh (2007)
16. Leidner, J.L.: An evaluation dataset for the toponym resolution task. Computers Environment and Urban Systems 30(4), 400–417 (2006)
17. Markowetz, A., Chen, Y., Suel, T.: Design and implementation of a geographic search engine. In: Proc. of WebDB, Baltimore, Maryland, pp. 19–24 (2005)
18. Silva, M.J., Martins, B.: Adding Geographic Scopes to Web Resources. Computers Environment and Urban Systems 30(4), 378–399 (2006)
19. Martins, B., Silva, M.J.: A Graph-Ranking Algorithm for Geo-Referencing Documents. In: Proc. If ICDM, Houston, Texas, pp. 741–744 (2005)
20. Wang, C., Xie, X., Wang, L., Lu, Y., Ma, W.: Detecting Geographic Locations from Web Resources. In: Proc. of GIR, Bremen, Germany, pp. 17–249
21. Sanderson, M., Kohler, J.: Analyzing geographic queries. In: Proc. of GIR, Sheffield, UK (2004)
22. Sanderson, M.: Retrieving with good sense. Information Retrieval 2(1), 45–65 (2000)
23. Sobhana, N., Barua, A., Das, M., Mitra, P., Ghosh, S.: Co-occurrence Based Place Name Disambiguation and its Application to Retrieval of Geological Text. In: Meghanathan, N., Boumerdassi, S., Chaki, N., Nagamalai, D. (eds.) NeCoM 2010, Part III. CCIS, vol. 90, pp. 543–552. Springer, Heidelberg (2010)
24. Volz, R., Kleb, J., Mueller, W.: Towards ontology-based disambiguation of geographical identifiers. In: Proc. of WWW Workshop on Identity, Identifiers, Identifications (I3), Bandd, Alberta, Canada (2007)

25. Zubizarreta, A., de la Fuente, P., Cantera, J.M., Arias, M.: Extracting geographic context from the Web: georeferencing in mymose. In: Proc. of GIR, pp. 554–561 (2009)
26. Wang, X., Zhang, Y., Chen, M., Lin, X.: An Evidence-based Approach for Toponym Disambiguation. In: Proc. of Geoinformatics 2010, pp. 1–7 (2010)
27. Wang, L., Wang, C., Xie, X., Forman, J., Lu, Y., Ma, W., Li, Y.: Detecting Dominant Locations from Search Queries. In: Proc. of SIGIR, Salvador, Brazil, pp. 424–431 (2005)
28. Rauch, E., Bukatin, M., Baker, K.: A confidence-based framework for disambiguating geographic terms. In: Proc. of HLT-NAACL-GEOREF, pp. 50–54 (2003)
29. Bryan, K., Leise, T.: The $25,000,000,000 Eigenvector: The Linear Algebra Behind Google. Journal SIAM Review 40(3), 569–581 (2006)