Title	On assigning place names to geography related web pages(conference paper)
Author(s)	Zong, Wenbo; Wu, Dan; Sun, Aixin; Lim, Ee Peng; Goh, Dion Hoe-Lian
Citation	Zong, W., Wu, D., Sun, A., Lim, E. P., & Goh, D. (2005). On Assigning Place Names to Geography Related Web Pages. Proceedings of the 5th ACM+IEEE Joint Conference on Digital Libraries JCDL 2005, (June 7-11, Denver, Colorado, USA), 354-362.
Date	2005
URL	http://hdl.handle.net/10220/6188
Rights	

Zong, W., Wu, D., Sun, A., Lim, E.P., and Goh, D.H. (2005). On assigning place names to geography related web pages. *Proceedings of the 5th ACM+IEEE Joint Conference on Digital Libraries JCDL 2005,* (June 7-11, Denver, Colorado, USA), 354-362.

On Assigning Place Names to Geography Related Web Pages

Wenbo Zong, Dan Wu, Aixin Sun, Ee-Peng Lim
Centre for Advanced Information Systems
School of Computer Engineering
Nanyang Technological University
Singapore, 639798
{aseplim }@ntu.edu.sg

Dion H. Goh
Division of Information Studies
School of Communication and Information
Nanyang Technological University
Singapore, 639798
ashlgoh@ntu.edu.sg

ABSTRACT

In this paper, we attempt to give spatial semantics to web pages by assigning them place names. The entire assignment task is divided into three sub-problems, namely place name extraction, place name disambiguation and place name assignment. We propose our approaches to address these sub-problems. In particular, we have modified GATE, a well-known named entity extraction software, to perform place name extraction using a US Census gazetteer. A rule-based place name disambiguation method and a place name assignment method capable of assigning place names to web page segments have also been proposed. We have evaluated our proposed disambiguation and assignment methods on a web page collection referenced by the DLESE metadata collection. The results returned by our methods are compared with manually disambiguated place names and place name assignment. It is shown that our proposed place name disambiguation method works well for geo/geo ambiguities. The preliminary results of our place name assignment method indicate promising results given the existence of geo/non-geo ambiguities among place names.

Keywords

1. INTRODUCTION

1.1 Motivation

An important research challenge in digital library systems is to provide the appropriate query capabilities to satisfy the information needs of users and applications. For many digital library collections, *query-by-location* works well when the objects to be retrieved can be specified using query predicates on the object locations. For example, a user planning for a holiday trip in Korea may want to find articles about Jeju, a popular resort island in the South of Korea. A student conducting beach erosion research may want to find documents about beaches in California and Florida. For these query examples, the complexity lies not in query evaluation, but in the extraction of location footprints from the documents. Unless tagged by creators or other users, it is not easy to determine the spatial coordinates or locations of documents. Besides query processing, place names appearing in documents can be used in applications such as:

- Providing the location information of events described by the documents[6];
- Enabling a map based visualization of documents[8]; and
- Mining spatial knowledge from documents or web pages containing both location and semantic concept information[10]. For example, one may want to find the cluster of web pages related to healthcare in Minnesota.

In the G-Portal digital library project, we treat geography related web pages and other types of web objects as as content resources and develop both map-based and classification-based interface to browse and query their metadata records[8]. As many of the web pages are not annotated with location when created, we would therefore like to address their *place name assignment problem*. Three main questions need to be addressed in the place name assignment problem: (a) what is the semantics of place name to be assigned to a document? (b) how can place names be identified and assigned? and (c) how can place name assignment be evaluated?

Given a web page, there are at least three place name semantics that can be adopted. The first place name semantics refers to the host location of the web page. This can be determined mainly by the domain name of the web page URL and can usually be carried out quite easily. There should only be one such location for each web page. The second semantics refers to the places described within web pages, e.g., a web page describing beaches in Hawaii. The third semantics refers to places as attributes of some events or objects, e.g., a web page describing a terrorist bombing event in Jarkata where Jarkata is the place attributes. Since our research deals with mainly geography related web pages, we adopt the second place name semantics. The identification of place names of the second and third semantics clearly requires content analysis. It is however noted that the extraction of place name attributes for events and objects is

usually covered in the *named entity extraction* tasks[2]. The extraction techniques for place names of the third semantics can also be used for place names of the second semantics but there are other issues to be addressed for the latter. Henceforth, unless otherwise stated, place names mentioned in the paper are of the second semantics.

In this paper, we aim to address the place name assignment task for for geography-related web pages. Several methods and systems have recently been developed for place name assignment as will be reviewed in Section 2. While most of them focus on the way place names are extracted and disambiguated, they have not explored how place names found in a web page can be used to determine the place names to be assigned to web pages and web page segments which are also known as the *page level* and *segment level place name assignment problems* respectively. Since a page is also a web page segment, we shall simply use place name assignment to mean both page and segment level place name assignments from now onwards. Indeed, place name assignment problem has to be solved in order to identify place names of the second semantics, and to filter away place names of the third semantics.

In place name assignment, the accuracies of extracting and disambiguating place names are clearly important. There are in addition two challenging research issues:

- A gazetteer consisting of place names is usually used to identify the place names occurring in a web page. Other than filtering away place names of third semantics, one has to assign place names at the appropriate granularity level as place names can be related to one another by containment (or parent-child) relationship, For example, New York city is part of New York state which is in turn part of USA.
- The goal is to accurately determine the segments where some places are the foci of description within them. An over-sized segment will be undesirable as it does not direct reader's focus to the most relevant part describing a place name. An under-sized segment, on the other hand, will miss parts of web page that are relevant.

1.2 Objectives and Contributions

In this paper, we define the place name assignment task as consisting of three subproblems, namely **place name extraction**, **place name disambiguation** and **place name assignment**. Place name extraction refers to identifying the place names appearing in web pages. The extracted place names provide the input to place name disambiguation. Disambiguation is necessary as each extracted place name may not have a unique match with some pre-specified dictionary of place names which is often known as a *gazetteer*. Without a unique match, the spatial location and type of the extracted place name cannot be determined. More details about place name disambiguation can be found in Section 4. In place name assignment, each web page or web page segment is assigned zero or more place names when the corresponding places are *significantly* described by the page.

We have chosen a collection of web pages referenced by DLESE project for this research[4]. The collection is chosen because many of these web pages contain place names[1] and we believe that the assigned place names will allow them to be spatially browsed and queried in G-Portal.

In the following, we summarize our contribution to the place name assignment task:

- *Place name extraction:* We use the well known GATE named entity extraction tool to extract place names from web pages. A new gazetteer containing mostly USA place names has been constructed and it allows GATE to easily extract place names from DLESE web pages.
- *Place name disambiguation :* A new place name disambiguation method based on heuristics rules has been developed. It consists of several steps each applying different set of rules to disambiguate place names. Our experiments have also shown very good disambiguation results.
- *Place name assignment:* A new place name assignment method for both page and segment levels has been proposed. The method incorporates the place name hierarchy in the given gazetteer to help assigning the most appropriate place name(s) to a web page or page segment.
- We have conducted some experiments to evaluate our disambiguation and place name assignment methods on randomly selected DLESE referenced web pages. A set of performance metrics have been defined. The results have been encouraging and we realised that geo/non-geo ambiguities adversely affected our place name assignment.

In our work, we assume no training dataset is given and human subject is used only during the evaluation phase. We also assume that a gazetteer consisting of place names organized with parent-child relationships is given.

1.3 Outline of Paper

The remaining sections of this paper are structured as follows. Section 2 gives an overview of the related research. Section 3 briefly describes the extraction of location names from web pages. Our proposed location disambiguation and location segment assignment methods are described in Sections 4 and 5 respectively. The experiments and results are given in Section 6. Finally, we give our conclusion in Section 7.

2. RELATED WORK

Place name assignment and its subproblems have been studied in several other research projects. Much of the previous research addresses the place name extraction and disambiguation sub-problems. Place name assignment and its variants, in contrast, have only been discussed in a few works[1, 11]. It has also been noted that there is generally a lack of reference corpus for research in this area[5].

[1]On average, 17 place names were extracted by GATE per page.

Manov et al. addressed place name extraction as part of the KIM project to construct a location knowledge base[9]. In the proposed approach, GATE is used to first extract place names with the help of a gazetteer consisting of 50,000 locations each with several aliases. Disambiguation of place names is performed using pattern-based grammar. In the Perseus project, proper names in historical documents are first identified using named entity extraction[12]. They are then matched against a gazetteer. The ambiguous ones are disambiguated by a series of heuristics based on the qualifiers in the vicinity (e.g., state name immediately following the city name), nearby disambiguated place names, and general world knowledge. Similar place name disambiguation method has been adopted in other research efforts[6].

In the system Web-a-Where, Amitay et al. addressed the place name extraction, disambiguation and page focus problems for web pages[1]. The page focus problem is similar to that of place name assignment except that the former is about assigning place names to web pages as a whole. Place names in a web page are identified by matching them with place names from a given gazatteer. Place names identified with multiple senses are disambiguated by confidence values derived from the qualifiers in the vicinity, human population, and disambiguating context that consists a set of place names to be disambiguated together. The proposed page focus strategy selects up to four place names that cover most of place names in a page. It is however not clear how the page focus strategy can be extended to handle place name problem.

MetaCarta is a commercial system that can perform place name extraction, disambiguation and place name-based query processing for web pages[11]. Again, a gazetteer is used for identifying place names in a web page. Each gazetteer place name is associated with a confidence value determined by the likelihood that it is correctly determined. NLP patterns, capitalization convention, place names found in vicinity, human population and other heuristics are further utilized to help disambiguating place names. To query web pages using a place name, they are scored by a function combining confidence values, positions and prominence of the place name in the web pages. MetaCarta does not address the place name problem at all.

3. PLACE NAME EXTRACTION

We extract place names from web pages using the GATE (or the main module ANNIE) software developed for extracting named entities. GATE has been chosen instead of implementing a different method because the former has been reported to give accurate place name extraction results[9]. Furthermore, our proposed place name assignment technique is designed to work with any place name extraction method. If necessary, it can always accommodate other extraction methods.

GATE consists of tokenizer, sentence splitter, POS tagger, and ontology matcher[2]. The types of named entities that can be extracted include person, location, organization, date, jobid, and money. In our implementation, only location is used.

For location entity extraction in GATE, a built-in gazetteer is used. The default gazetteer consists of 6713 place names from different countries and place names appearing in a web page will be directly identified by matching them against the gazetteer. GATE also applies some natural language and linguistic patterns to identify place names that may not appear in the gazetteer.

As a large portion of web pages referenced by DLESE project are about USA, and the GATE's default gazetteer has insufficient information about detailed USA locations, we decided to incorporate the US Census 2000 gazetteer[13] into GATE so that US place names can be extracted with better accuracy. The new gazetteer also facilitates the construction of a hierarchical view of US places that can be used in disambiguating the extracted place names and conducting place name assignment.

Table 1 presents the statistics of the Census 2000 gazetteer. Place names are divided into four granularity levels: *state*, *city*, *county*, and *county subdivision*. Both cities and counties are grouped under states and county subdivisions are grouped under counties. Each place name is associated with a spatial point location. As place names in Census gazetteer are often appended with common suffixes (e.g., city, town), we produced aliases for some place names by removing the common suffixes and included the aliases into the gazetteer. This is based on the observation that users often refer to a place without explicitly stating whether it is a state, city or others. For instance, "Carbon Hill" was included as an alias for "Carbon Hill city".

Table 1: Statistics of Gazetteer

Place Name Type	Number
State	52
City	25,375
County	3219
County subdivision	36,351

4. PLACE NAME DISAMBIGUATION

Exact string matching against place names in the gazetteer and applying other extraction patterns result in a set of place names that could be ambiguous. Amitay et al. defines two types of place name ambiguities, namely *geo/non-geo* and *geo/geo*[1]. The former refers to ambiguities between a geographic place name and a non-geographic place name. For example, "Washington" could be a state name or a person name, and "Welcome" could be city name[3] or a common English word. Geo/geo ambiguities are those that involve a pair of geographic place names that are similar. For example, "New York" is a name for both state and city in USA. "Cambridge" is a city in both USA and UK. As an extreme example, there are more than five "Washington county" in USA. As GATE has incorporated some extraction patterns to handle geo/non-geo ambiguities but not the geo/geo ambiguities[3], we mainly focus on resolving latter using a set of disambiguation rules.

[2]GATE can be downloaded from http://www.gate.ac.uk. More detail can be found in [3].

[3]There is a Welcome City in Minnesota, USA.

Our rule based disambiguation approach makes use of both *contextual information* extracted from the web pages and *spatial distances* between place names. Contextual information consists of *self-features* and *near context*. Self-features refer to features derived by applying some patterns relating a place name with some place sense. The near context refers to small set of of words that appear after the given place name. Spatial distance between two place names denoted by $dist(p_1, p_2)$ refers to distance between them. Distance between two states is measured by the shortest distance among all pairs of cities from the two states. Distance between a non-state place and a state is defined as the shortnest distance between a place within the state and the non-state place.

The following outlines our disambiguation algorithm.

Step 0: (Initialization)

1. Let P_{am} be the set of $\langle p_i, doc\_pos_i, P_i^g \rangle$ tuples, where p_i is a place name to be disambiguated, doc_pos_i refers to its position in the given web page, and P_i^g refers to the set of place names p_{ij}^g from the gazetteer that can possibly be matched with p_i.
2. Let P_{da} be the set of $\langle p_i, doc\_pos_i, p_i^g \rangle$ tuples, where p_i is a place name already disambiguated, doc_pos_i refers to its position in the given web page, and p_i^g refers to the disambiguated place name from the gazetteer.
3. P_{da} is initialized to be empty.

Step 1: (Self-Feature Extraction)

1. For each $\langle p_i, doc\_pos_i, P_i^g \rangle \in P_{am}$ such that one of the following *self-feature patterns* applies:
   (a) p_i + "," + place name sense (e.g., *Chicago*, an old city)
   (b) Place name sense + "of" + p_i (e.g., state of *Califormia*)
   (c) p_i + place name sense (e.g., *Rio Grande* County)

   Remove p_{ij}^g's that violate the place name sense from P_i^g.
   If ($|P_i^g| == 1$), add $\langle p_i, doc\_pos_i, P_i^g \rangle$ to P_{da}.
2. For each $\langle p_i, doc\_pos_i, P_i^g \rangle \in P_{am}$ such that a state name appears in the *near context* of p_i (e.g. Philadelphia is in PA.):

   Remove p_{ij}^g's that are not within the state.
   If ($|P_i^g| == 1$), add $\langle p_i, doc\_pos_i, P_i^g \rangle$ to P_{da}.

Step 2: (Perfect Matching)

1. If P_{da} is empty

   For each $\langle p_i, doc\_pos_i, P_i^g \rangle \in P_{am}$

   if there exists exactly one perfect match between a place name p_{ij}^g in gazetteer and p_i

   add $\langle p_i, doc\_pos_i, \{p_{ij}^g\} \rangle$ to P_{da}
   remove $\langle p_i, doc\_pos_i, P_i^g \rangle$ from P_{am}

Step 3: (Propagation of Disambiguated Place Name Senses)

1. While P_{da} is growing

   For each $\langle p_i, doc\_pos_i, P_i^g \rangle \in P_{am}$

   Let p be the most adjacent disambiguated place name of p_i
   If p and p_i are involved in one of the following patterns:
   (a) p_i + "," + p where p has the state sense (e.g., "Denver, Colorado")
   (b) p_i + "of" + p where p has the state sense (e.g., "Berkeley of California")
   (c) p_i + p where p has the state sense (e.g., "Buffalo NY")
   (d) p_i + "|" + p where p has the state or city sense (e.g., "Wisconsin | Minnesota")
   (e) p + "|" + p_i where p has the state or city sense

   Remove p_{ij}^g's from P_i^g if the former does not belong to the state p for cases (a) to (c), or if the former does not have the matching state or city senses for cases (d) and (e).
   If ($|P_i^g| == 1$), add $\langle p_i, doc\_pos_i, P_i^g \rangle$ to P_{da} and remove it from P_{am}.
2. For each $\langle p_i, doc\_pos_i, P_i^g \rangle \in P_{am}$

   if there exists exactly one perfect match between a place name p_{ij}^g in gazetteer and p_i

   add $\langle p_i, doc\_pos_i, \{p_{ij}^g\} \rangle$ to P_{da}
   remove $\langle p_i, doc\_pos_i, P_i^g \rangle$ from P_{am}

Step 4: (Spatial Distance-based Disambiguation)

1. For each $\langle p_i, doc\_pos_i, P_i^g \rangle \in P_{am}$

   Let p be the most adjacent disambiguated place name of p_i
   Let $p_{ik}^g = arg\ min_{p_{ij}^g \in P_i^g}(dist(p, p_{ij}^g))$
   Add $\langle p_i, doc\_pos_i, \{p_{ik}^g\} \rangle$ to P_{da}
   Remove $\langle p_i, doc\_pos_i, P_i^g \rangle$ from P_{am}

In our algorithm, we start with an ambiguous set of place names in P_{am} and we want to construct the disambiguated set of place names in P_{da}.

In Step 1, self features of each ambiguous place name are derived by a set of self-feature patterns. In the Census gazetteer, it is easy to tell the place name sense as all place names (except state names) carry suffixes that explicitly states the place name senses. In addition, if a state abbreviation occurs in the near context of the place name, the

place name is assumed to belong to that state. In our experiment, near context consists of three words that come after the place name and this appeared to work well for our dataset. Both self features and near context can eliminate some options for the ambiguous place names. Some place names may be completely disambiguated and added to P_{da}.

Step 2 is required when no place names can be completely disambiguated in Step 1. This step determines those ambiguous place names that can perfectly match some place names (in the gazetteer) uniquely, and disambiguates them accordingly.

In Step 3, the place name senses of the disambiguated place names are propagated to help disambiguating other place names. This is achieved by employing patterns involving some disambiguated place name sense and an ambiguous place name. Patterns (a) to (c) imply a containment relationship between place names and state sense, whereas patterns (d) and (e) indicate some serial relationship for states or cities. These patterns are defined empirically. Patterns are applied to the ambiguous place names as long as new places are successfully disambiguated. Only after no new place name can be disambiguated using patterns, perfect matching is applied to disambiguate ambiguous ones that have been left with only one perfect matching place. This is necessary because some places have more than one perfect matching place in the gazetteer(e.g., there are a dozen of Washington city's in US), and some of the candidates may not be eliminated until after patterns are applied.

In Step 4, we compute for each ambiguous place name the spatial distance ($dist()$) between its every disambiguation option and the most adjacent disambiguated place name. The option that yields the minimum distance will be used to disambiguate the place name.

Some other methods[7] that use default sense of place names to disambiguate place names. Default sense usually refers to choosing among the disambiguation place name options the one that is more popular or most populated. Default sense is not used in our approach as it may not be correct for some place names that are meant for less popular or populated locations. Instead, our technique uses the self-features and near context to obtain more localized information about the ambiguous place names. Our technique also does not assume that the same sense be used for all occurrences of a place within a web page. Each occurrence is treated separately to avoid propagating the wrong decisions.

5. PLACE NAME ASSIGNMENT

Recall that the objective of place name assignment is to determine for a web page or page segment a set of places it describes. With place names assigned, web pages can be easily accessed by place names and even their spatial locations.

Place name assignment however is not a simple problem. A simple method of counting place name occurrences in the web page and assigning the ones with frequencies higher than some threshold does not necessarily perform well. Firstly, this method does not consider the hierarchical relationships between place names. When several child place names of a common parent place name occur frequently, it will be more logical to assign the parent place name instead of the individual child place names. For example, if China, Korea and Japan frequently appear in a page, it is probably better to assign East Asia to the page. In some cases, the parent place name may not even appear in the page. Secondly, it is difficult to determine the count threshold that does not bias against small web pages. A normalized count can be adopted instead but this becomes complicated when the hierarchical relationships between place names are considered. Moreover, all place names appearing in a web page should be correctly identified before place name assignment but this is difficult to achieve in practice.

In this section, we describe our proposed place name assignment method. It can handles both assignment to pages and assignment to segments. It incorporates the normalized counts of place name occurrences in the web page, and derives the normalized count of a parent place name using those of its child place names.

5.1 Algorithm Outline

Our proposed place name assignment algorithm takes a web page and its disambiguated place names as input, constructs a segment tree for the page, and assigns appropriate place names to each segment in the tree. Algorithm 1 outlines the steps in assigning place names to a web page D.

Algorithm 1 Assign_Place_Name(G, D, L)

Input: gazetteer G,
web page D,
disambiguated place set L
output: assigned place names P with respective segments

1: initialize P := Perform depth-first-traversal on D to find segments tree S such that each leaf in S is a minimal subtree in D containing $\geq n_s$ terms
2: **for** each $s \in S$ **do**
3: Construct a gazetteer subtree T_s of G such that $T_s = \{p | p$ appears in s or p is an ancestor place name of some p' in $s\}$
4: **for** each $p \in T_s$ in bottom-up order **do**
5: $count(s,p)$ = number of occurrences of p in s
6: $count(s)$ = number of terms in s
7: **if** p is a leaf place name in T **then**
$score(s,p) = w_{parent} \cdot \frac{count(s,p)}{count(s)}$
8: **else**
$P_c = \{p_c | p_c$ is a child place name of $p\}$
$children\_score(s,p) = \sum_{p_c \in P_c} score(s,p_c)$
$f(s,p) = \sum_{p_c \in P_c}(-score(s,p_c)/children\_score(s,p) \cdot \log_2(score(s,p_c)/children\_score(s,p)))$
$score(s,p) = (w_{parent} \cdot \frac{count(s,p)}{count(s)}) + (w_{child} \cdot f(s,p) \cdot children\_score(s,p))$
9: **end if**
10: **end for**
11: **end for**
12: **Return** $\{(s,p) | s \in S, p \in T_s, score(s,p) \geq \alpha_{score}\}$

As a web page is essentially a document tree, the algorithm

traverses the tree to construct segments and the segment tree in a depth-first fashion. A new segment is created when a document node (or *tag*, in HTML term) contains more than n_s terms ($n_s = 200$ in our experiments). This serves to reduce the number of segments and also eliminate segments that are too small for processing. A parent segment containing one or more child segments is represented by a set of parent-child links in the segment tree. An example segment tree is illustrated in Figure 1.

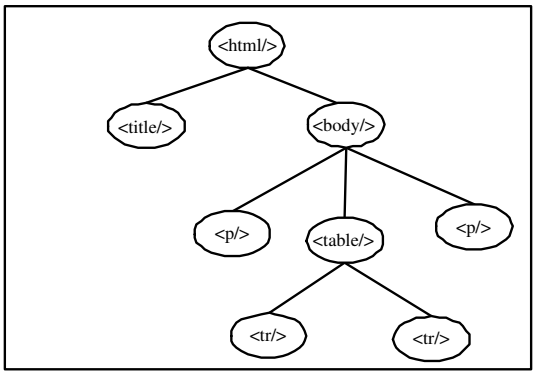

Figure 1: Segment tree

For each segment, a subtree of the US gazetteer hierarchy is constructed to cover all place names appearing in the segment and their ancestor place names. We call this the *gazetteer subtree*. It should be noted that the ancestor place names may not appear in the segment. This will allow them to be assigned to the segment if the child place names appear frequently enough.

Given a segment, a score is computed for each place name in the gazetteer subtree, indicating its relevance to the segment. For the leaf place names in the gazetteer subtree, the score is simply the number of its occurrences divided by total number of terms in the segment weighted by a factor w_{parent}. The two weight factors, w_{parent} and w_{child}, with $w_{parent} + w_{child} = 1$ are used to determine the contributions of score from a place name and all its child place names respectively.For a non-leaf place name, the score contribution from its child place names includes:

- Total score of the child place names: that measures how relevant its children are to the segment, and
- Distribution of scores among the child place names: that measures how evenly its children's scores are distributed.

The latter is important because it measures the degree to which they *together* contribute to the parent place. That is, the more evenly their scores are distributed, the more they contribute to the parent. This is illustrated in the following example. If Los Angeles is the only city in California mentioned in a segment, we then should consider assigning Los Angeles rather than California because a more specific place is preferred to a more general one. On the other hand, if Los Angeles, San Francisco, San Diego and Sacramento appears evenly in the segment, we should consider assigning California to the segment as it is more likely the place name described by the segment than any individual city. In the extreme case, suppose Sacramento appears many times in the segment while the other three cities appear scarcely, our method is designed to assign only Sacramento to the segment and drop the other three because their existence is relatively insignificant. If California also does not appear frequently enough in the segment, it will not gain much support from Sacramento since the scores of the four cities are not even. Consequently, the score of California should be small and should not be assigned to the segment.

The distribution of children's scores of a place name p in a segment s is denoted by $f(s,p)$ in Algorithm 1. The formula measures the entropy or the degree of randomness arising from the scores of child place names. Let P_c denote the set of child place names of a parent place name p, and $children\_score(s,p)$ denote the total score of child place names of p, $f(s,p)$ is defined as follows:

$$f(s,p) = \sum_{p_c \in P_c}(-score(s,p_c)/children\_score(s,p) \cdot \log_2(score(s,p_c)/children\_score(s,p)))$$

In the above formula, the more evenly the child place names are distributed, the higher $f(s,p)$ is. The following example shows how $f(s,p)$ is computed and how it affects the final score of the parent place.

Consider a segment with 200 terms and two occurrences of California. Assume that four cities in California, namely Los Angeles, San Francisco, San Diego and Sacramento, appear four times each, giving total occurrences of 16 times. The score for each of the four cities is hence $0.7 \cdot \frac{4}{200} = 0.014$, and the score for California is computed as follows:

- $children\_score(s,p) = 4 \cdot 0.014 = 0.056$
- $f(s,p) = 4 \cdot (-0.014/0.056 \cdot \log_2(0.014/0.056)) = 2$
- $score_{California} = 0.7 \cdot \frac{2}{200} + 0.3 \cdot 2 \cdot 0.056 = 0.0406$

Now assume that Los Angeles, San Francisco and San Diego each appear only once, and Sacramento appears 13 times, again giving total occurrences of 16 times. The score for each of the first three cities is then given by $0.7 \cdot \frac{1}{200} = 0.0035$, and that for Sacramento is given by $0.7 \cdot \frac{13}{200} = 0.0455$. The score for California is now computed as follows:

- $children\_score(s,p) = 3 \cdot 0.0035 + 0.0455 = 0.056$
- $f(s,p) = 3 \cdot (-0.0035/0.056 \cdot \log_2(0.0035/0.056)) + (-0.0455/0.056 \cdot \log_2(0.0455/0.056)) = 0.9934$
- $score_{California} = 0.7 \cdot \frac{2}{200} + 0.3 \cdot 0.9934 \cdot 0.056 = 0.0237$

Comparing the scores for California in the above two cases shows that the distribution of children's scores greatly affect their contribution to the parent, according to our algorithm. Note that California always occurs twice and four cities together occur 16 times. When the cities appear equal number

of times, the final score for California is significantly larger than when one of them appears many more times than the others (0.0406 vs. 0.0237).

Once the scores of place names in the gazetteer subtree are derived, those place names with scores greater or equal to α_{score} will be assigned to the segment. The threshold α_{score} can be chosen based on the type of web page collection. A higher threshold can be used if the pages are known to be geography-related. Otherwise, a lower threshold can be used for web pages of general content. Empirically, α_{score} can be determined by taking a small set of sample pages as training data.

The output of the place name assignment algorithm is a set of pairs (s, p) indicating that segment s is assigned with place p. Note that a segment s could be at any level and there could be segments that are not assigned with any place name at all.

6. EXPERIMENTAL RESULTS

In this section, we describe the experiments conducted to evaluate the accuracy of our place name disambiguation method and place name assignment method on a collection of web pages created from the DLESE metadata collection[4]. For place name assignment, we first introduce the evaluation metrics for both page-level assignment and segment-level assignment. Due to time constraint, we conducted the evaluation on a set of 50 web pages randomly chosen for manual checking for page-level assignment. The same was done for segment-level assignment. As only a US gazetteer was used, in our experiments, we focused only on US place names.

6.1 Dataset

In our experiment, the DLESE dataset was created by downloading web pages referenced by DLESE metadata records. DLESE is an ongoing NSDL digital library project that gathers metadata of earth science related web objects including web sites, web pages and other types of files. Thirty concurrent crawler threads were used and they were programmed to skip files with extensions *.doc, .gif, .jpg, .mov, .mpg., pdf, .xml* and *.ppt*. Most of the downloaded web pages have extensions *html, htm* and *txt*. As shown in Table 2, a total of 8726 web pages were finally included in the DLESE dataset, and they are referenced by 8835 metadata records[4]. Note that there could be more web pages downloaded by following the links in these DLESE web pages. As their relevance to geography content cannot be easily determined, we have chosen not to include these indirectly referenced web pages. Table 3 lists the top 5 web sites referenced by DLESE metadata records. They together contribute more than 30% of the web pages. This information may be useful if site specific semantics can be later incorporated to handle web pages of these popular web sites.

6.2 Evaluation of Disambiguation Method

As there are 8835 web pages directly referenced by DLESE metadata records and place names in them have not been manually labelled, we have randomly chosen 50 pages containing more than 31 and less than 200 occurrences of place names from the collection and evaluate our place name disambiguation method on them. The place names of these web pages were first extracted using the extraction method described in Section 3.

[4]The same web page may be referenced by multiple metadata records.

Table 2: Overview of DLESE Dataset

Total number of DLESE metadata records used	8835
Total number of distinct URLs	8726
URLs referenced by multiple resources	109
Number of distinct web sites	2218

Table 3: Top 5 Web hosts

Web host (Web site)	Resources
svs.gsfc.nasa.gov	2091
www.nationalgeographic.com	168
www2.nature.nps.gov	129
www.ucmp.berkeley.edu	128
serc.carleton.edu	116

In place name disambiguation, a place name can either be correctly or wrongly disambiguated. For the wrongly disambiguated place names, we consider two types of errors: geo/geo and geo/non-geo. A geo/geo error refers to a case where the extracted named entity is a place name but an incorrect place name is assigned. A geo/non-geo error refers to a case where the extracted named entity is in fact not a place name, but has been assigned a place name during disambiguation. Unfortunately, geo/non-geo errors cannot be recognized in our disambiguation method as our method assumes that the named entities extracted from GATE are some place names.

For the 50 randomly chosen web pages in our experiment, after eliminating the 1185 non-US named entities[5] (no possible matches with US places), there was a total of 1387 named entities, and 760 of them were place names. Among the 760 extracted place names, we found that 675 have been correctly disambiguated by our method, giving a precision of 88.8%. This was done by manually checking the 760 disambiguated place names. Among the 675 correctly disambiguated place names, the contributions of different heuristic rules are tabulated in Table 4. The result shows that Perfect Matching is the most effective rule in out experiment. Spatial Distance-based heuristic rule also plays an important role in disambiguation, contributing more than one third to correct disambiguation. Propagation of the disambiguated place name senses does not help too much, probably due to the fact that it is carried out after the Self-Feature rule is applied.

6.3 Evaluation Metrics For Place Name Assignment

We have adopted two metrics for evaluating the performance of our place name assignment algorithm. The first metric,

[5]Recall that we added more entries to GATE gazetteer, did not replace it

Table 4: Contribution of heuristic rules

Rule	Places disambiguated	Percentage
Self-Feature Extraction	96	12.6%
Perfect Matching	355	46.7%
Propagation of place name senses	31	4.1%
Spatial Distance-based	278	36.6%

page-centric accuracy, is for evaluating the performance of assigning place names to the pages in page level place name assignment. The second metric, known as *place-centric accuracy*, measures the degree of accuracy of assigning a place to a particular segment in segment level place name assignment.

For the page level place name assignment, a human subject is given a set of pages and a place assigned to each of them. He or she is expected to give one of two possible responses at his/her discretion for each page:

A: The place name assignment is correct for the page.

B: The place name assignment is not correct for the page.

The page-centric accuracy is thus defined by

$$\text{Page-centric accuracy} = \frac{\#A}{\#A+\#B}$$

To measure the performance of segment level place name assignment, a human subject is given a segment s and a place name p assigned to it. He or she is expected to tell to what degree the segment can be assigned with the given place. One of four possible responses will be given at his/her discretion for the segment-place name pair:

C: The assignment of p to s is completely wrong - For example, p and its child place names does not appear in s at all.

D: The segment s is too large to be assigned with p - In other words, p is relevant to s but the given s covers other segments that should be excluded.

E: The segment s is too small to be assigned with p - Similar to D except that s is too small now.

F: The segment s is just about the right region to be assigned with p - This is the most ideal situation indicating that the assignment is good.

It should be noted that responses D and E naturally involve some degree of tolerance, subject to the human subject. If the segment s is way too large or too small to be assigned with place p, the human subject should respond with C instead. That is, the assignment should be considered as wrong if it is beyond a certain degree of tolerance. We further define two accuracy measures for place-centric metric as shown below:

$$\text{Hard accuracy} = \frac{\#F}{\#C+\#D+\#E+\#F}$$
$$\text{Relaxed accuracy} = \frac{\#D+\#E+\#F}{\#C+\#D+\#E+\#F}$$

Obviously, *relaxed accuracy* is greater than or equal to *hard accuracy*, as it relaxes the criteria for the "right" region. Relaxed accuracy is designed to give more weight to correct place name than the correct segment.

6.4 Experimental Results For Place Name Assignment

In the experiment for both page level and segment level place name assignments, we assigned the best place name to each web page. This is achieved by assigning for each page the place name with highest score computed by Algorithm 1. This effectively removed the need for the threshold α_{score} in this evaluation.

For page level place name assignment, a random sample of 50 pages were chosen. A human subject was then asked to give his/her responses to each of these pages and their page name assignment. As depicted in Table 5, 33 out of 50 page level place name assignments were considered correct, and 17 were incorrect. This gave a page-centric accuracy of 66%. When the incorrect assignments were examined further, it was found that 12 of them were in fact due to geo/non-geo errors during disambiguation. As our assignment method does not really deal with geo/non-geo ambiguities. it did not perform well. We therefore evaluated the performance of our place name assignment algorithm again with geo/non-geo errors discarded and obtained a page-centric accuracy of $\frac{33}{50-12} = 86.8\%$.

Table 5: Performance for page level assignment

Response	Entity category	Number	Percentage
A	geo/geo	33	66%
B	geo/geo	5	10%
	geo/non-geo	12	24%

For evaluating the accuracy of segment level assignment, another random sample of 50 pages was chosen, with each page having the place with highest score assigned to a segment and the segment-place pair identified. A human subject was given these pages and asked to give his/her responses to each of the segment-place pairs. As shown in Table 6, assignment at segment level for 29 pages was considered totally wrong, 3 pages with segment being too large, 3 pages with segment being too small, and 15 pages with segment being just right. Again, further inspection shows that 27 out of the 29 pages were due to geo/non-geo errors, and 2 of them were actually assigned with wrong places. Discarding pages with geo/non-geo errors, the *hard accuracy* for the place-centric metric is $\frac{15}{50-27} = 65.2\%$, *relaxed accuracy* $\frac{3+3+15}{50-27} = 91.3\%$.

The hard accuracy for the place-centric metric turns out to be a very strict measure. Not only does it require the correct assignment of a place with high confidence, it also requires the "right" region to be assigned together with the place. In fact, deciding the "right" region is more subjective than deciding the correct place as it depends on the organization of the content of a page in addition to readers' discretion. In our experiments, we have only considered the segment

Table 6: Performance for segment level assignment

Response	Entity category	Number	Percentage
C	geo/geo	2	4%
	geo/non-geo	27	54%
D	geo/geo	3	6%
E	geo/geo	3	6%
F	geo/geo	15	30%

assigned with a place with the highest score across the whole page. It could be the case that a particular assignment can apply to a larger region but with a reduced confidence, which is what response E indicates. On the other hand, it could be the case that related places are mentioned in some parts of the segment, all contributing some score to the assigned place, but there is one portion mentioning the assigned place more intensively. This may lead readers to conclude that the place is assigned a region that is too large, which is what response D indicates. It should be noted that in the page-centric metric, we do not consider such cases, and the focus is on whether the assignment of a place to a page makes sense. By relaxing the "right" region criterion, the relaxed accuracy value (91.3%) is consistent with the page-centric accuracy value (86.8%).

It should also be noted that most wrong assignments both at page level and at segment level are due to geo/non-geo errors, which is not handled by our place name assignment method. It hence suggests that to achieve better assignment accuracy, geo/non-geo errors have to be eliminated at the place name extraction and disambiguation phases.

7. CONCLUSION

This paper describes the task of assigning place names to geography-related web pages so as to discover more semantics about the web pages. We focus on two important sub-problems in the task, i.e., place name disambiguation and place name assignment. We devise a rule-based disambiguation method that uses self-features, near context, perfect match, extraction patterns, and spatial distance to systematically determine a gazetteer place name to each ambiguous place name. The method achieved good precision for a random sample of 50 web pages referenced by DLESE metadata records. We also propose a place name assignment method that considers the contributions of child place names during the assignment. This assignment method has been evaluated using both page-centric and place-centric accuracies designed for evaluating page level and segment level place name assignments respectively. Although we have obtained some interesting results, much more research remains to be investigated and we highlight two below:

- There are more extensive experiments that should be conducted to evaluate our proposed place name disambiguation and place name assignment methods. Due to time constraint, the current evaluation has been done on a small set of web pages and has not included other possible methods.
- We have noted that geo/non-geo ambiguities contributed significantly to the errors made by our methods. We therefore plan to extend our place name disambiguation method to handle them.

8. ACKNOWLEDGMENTS

This work is partially funded by the Centre for Research in Pedagogy and Practice, National Institute of Education through the Project No. CRP 40/03 LEP. We also wish to acknowledge the contribution by John Hedberg, Chew-Hung Chang and Yin-Leng Theng for their useful comments.

9. REFERENCES

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: Geotagging web content. In *SIGIR 2004*, Sheffield, South Yorkshire, UK, July 2004.

[2] N. Chinchor. Muc-7 named entity task definition version 3.5. In *Seventh Message Understanding Conference (MUC-7)*, 1998.

[3] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[4] Digital Library for Earth System Education. http://www.dlese.org.

[5] J. Leidner. Towards a reference corpus for automatic toponym resolution evaluation. In *SIGIR 2004*, Sheffield, South Yorkshire, UK, July 2004.

[6] H. Li, R. Srihari, C. Niu, and W. Li. Location normalization for information extraction. In *19th Conference on Computational Linguistics (COLING'02)*, Taipei, Taiwan, August 2002.

[7] H. Li, R. K. Srihari, C. Niu, and W. Li. Infoxtract location normalization: a hybrid approach to geographic references in information extraction. In *Proc. of HLT-NAACL 2003 Workshop on Analysis of Geographic References*, Alberta, Canada, 2003.

[8] E.-P. Lim, D. H.-L. Goh, Z. Liu, W.-K. Ng, C. S.-G. Khoo, and S. E. Higgins. G-portal: A map-based digital library for distributed geospatial and georeferenced resources. In *Proceedings of the Second ACM+IEEE Joint Conference on Digital Libraries (JCDL 2002)*, Portland, Oregon, USA, July 14-18 2002.

[9] D. Manov, A. Kiryakov, B. Popov, K. Bontcheva, and D. Maynard. Experiments with geographic knowledge for information extraction. In *HLT-NAACL 2003 Workshop on Analysis of Geographic References*, Edmonton, Canada, 2003.

[10] Y. Morimoto, M. Aono, M. E. Houle, and K. McCurley. Extracting spatial knowledge from the web. In *Symposium on Applications and the Internet (SAINT'03)*, 2003.

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

[12] D. Smith and G. Crane. Disambiguating geographic names in a historical digital library. In *ECDL*, pages 127–136, 2001.

[13] US Census Bureau. http://www.census.gov.