# Location Normalization for Information Extraction*

**Huifeng Li, Rohini K. Srihari, Cheng Niu, and Wei Li**
Cymfony Inc.
600 Essjay Road, Williamsville, NY 14221, USA
(hli, rohini, cniu, wei)@cymfony.com

## Abstract

Ambiguity is very high for location names. For example, there are 23 cities named 'Buffalo' in the U.S. Country names such as 'Canada', 'Brazil' and 'China' are also city names in the USA. Almost every city has a Main Street or Broadway. Such ambiguity needs to be handled before we can refer to location names for visualization of related extracted events. This paper presents a hybrid approach for location normalization which combines (i) lexical grammar driven by local context constraints, (ii) graph search for maximum spanning tree and (iii) integration of semi-automatically derived default senses. The focus is on resolving ambiguities for the following types of location names: island, town, city, province, and country. The results are promising with 93.8% accuracy on our test collections.

## 1 Introduction

The task of location normalization is to identify the correct sense of a possibly ambiguous location Named Entity (NE). Ambiguity is very serious for location NEs. For example, there are 23 cities named 'Buffalo', including the city in New York State and in Alabama State. Even country names such as 'Canada', 'Brazil', and 'China' are also city names in the USA. Almost every city has a Main Street or Broadway. Such ambiguity needs to be properly handled before converting location names into some normal form to support entity profile construction, event merging and visualization of extracted events on a map for an Information Extraction (IE) System.

Location normalization is a special application of word sense disambiguation (WSD). There is considerable research on WSD. Knowledge-based work, such as (Hirst, 1987; McRoy, 1992; Ng and Lee, 1996) used hand-coded rules or supervised machine learning based on annotated corpus to perform WSD. Recent work emphasizes corpus-based unsupervised approach (Dagon and Itai, 1994; Yarowsky, 1992; Yarowsky, 1995) that avoids the need for costly truthed training data. Location normalization is different from general WSD in that the selection restriction often used for WSD in many cases is not sufficient to distinguish the correct sense from the other candidates.

For example, in the sentence "The White House is located in Washington", the selection restriction from the collocation 'located in' can only determine that "Washington" should be a location name, but is not sufficient to decide the actual sense of this location. Location normalization depends heavily on co-occurrence constraints of geographically related location entities mentioned in the same discourse. For example, if 'Buffalo', 'Albany' and 'Rochester' are mentioned in the same document, the most probable senses of 'Buffalo', 'Albany' and 'Rochester' should refer to the cities in New York State. There are certain fixed keyword-driven patterns from the local context, which decide the sense of location NEs. These patterns use keywords such as 'city', 'town', 'province', 'on', 'in' or other location names. For example, the pattern "X + city" can determine sense tags for cases like "New York city"; and the pattern "City + comma + State" can disambiguate cases such as "Albany, New York" and "Shanghai, Illinois". In the absence of these patterns, co-occurring location NEs in the same discourse can be good evidence for predicting the most probable sense of a location name.

*This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-00-C-0090.

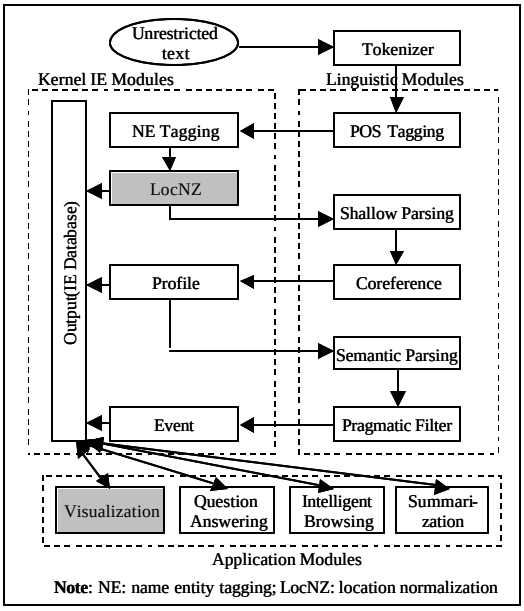


Figure 1. InfoXtract system architecture

For choosing the best matching sense set within a document, we simply construct a graph where each node represents a sense of a location NE, and each edge represents the relationship between two location name senses. A graph spanning algorithm can be used to select the best senses from the graph. If there exist nodes that cannot be resolved in this step, we will apply default location senses that were extracted semi-automatically by statistical processing. The location normalization module, or 'LocNZ', is applied after the NE tagging module in our *InfoXtract* IE system as shown in Figure 1.

This paper focuses on how to resolve ambiguity for the names of island, town, city, province, and country. Three applications of LocNZ in Information Extraction are illustrated in Section 2. Section 3 presents location sense identification using local context; Section 4 describes disambiguation process using information within a document through graph processing; Section 5 shows how to semi-automatically collect default senses of locations from a corpus; Section 6 presents an algorithm for location normalization with experimental results. The summary and conclusions are given in Section 7. Sample text and the results of location tagging are given in the Appendix.

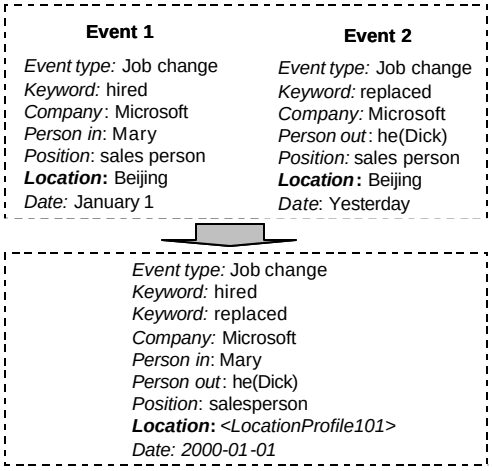


Figure 2. Location verification in Event merging.

## 2 Applications of Location Normalization

Several applications are enabled through location normalization.

- Event extraction and merging

Event extraction is an advanced IE task. Extracted events can be merged to provide key content in a document. The merging process consists of several steps including checking information compatibility such as checking synonyms, name aliases and co-reference of anaphors, time and location normalization. Two events cannot be merged if there is a conflicting condition such as time and location. Figure 2 shows an example of event merging where the events occurred in Microsoft at Beijing, not in Seattle.

- Event visualization

Visualization applications can illustrate where an event occurred with support of location normalization. Figure 3 demonstrates a visualized event on a map based on the normalized location names associated with the events. The input to visualization consists of extracted events from a news story pertaining to Julian Hill's life. The arrow points to the city where the event occurred.

- Entity profile construction

An entity profile is an information object for entities such as person, organization and location. It is defined as an Attribute Value Matrix (AVM) to represent key aspects of information about entities, including their relationships with other entities. Each attribute slot embodies some

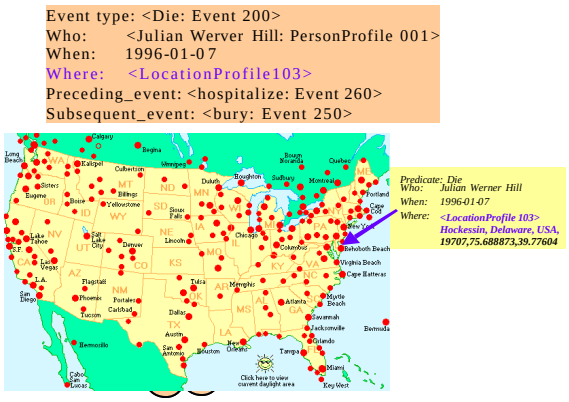


Figure 3. Event visulization with location.

information about the entity in one aspect. Each relationship is represented by an attribute slot in the Profile AVM. Sample Profile AVMs involving the reference of locations are illustrated below.

*<PersonProfile 001> ::*
Name: Julian Werner Hill
Position: Research chemist
Age: 91
*Birth-place*: <LocationProfile100>
Affiliation: Du Pont Co.
Education: MIT

*<LocationProfile 100> ::*
Name: St. Louis
State: Missouri
Country: United States of America
Zipcode: 63101
Lattitude: 90.191313
Longitude: 38.634616
Related_profiles: <PersonProfile 001>

Several other applications such as question answering and classifying documents by location areas can also be enabled through LocNZ.

## 3 Lexical Grammar Processing in Local Context

Named Entity tagging systems (Krupka and Hausman, 1998; Srihari et al., 2000) attempt to tag information such as names of people, organizations, locations, time, etc. in running text. In InfoXtract, we combine Maximum Entropy Model (MaxEnt) and Hidden Markov Model for NE tagging (Shrihari et al., 2000). The Maximum Entropy Models incorporate local contextual evidence in handling ambiguity of information from a location gazetteer. In the Tipster Location gazetteer used by InfoXtract, there are a lot of common words, such as *I, A, June, Friendship*, etc. Also, there is large overlap between person names and location names, such as *Clinton, Jordan*, etc. Using MaxEnt, systems learn under what situation a word is a location name, but it is very difficult to determine the correct sense of an ambiguous location name. If a word can represent a city or state at the same time, such as *New York* or *Washington*, it is difficult to decide if it refers to city or state. The NE tagger in InfoXtract only assigns the location super-type tag NeLOC to the identified location words and leaves the task of location sub-type tagging such as NeCITY or NeSTATE and its normalization to the subsequent module LocNZ. For representation of LocNZ results, we add an unique zip code and position information that is longitude and latitude for the cities for event visualization.

The first step of LocNZ is to use local context that is the co-occurring words around a location name. Local context can be a reliable source in deciding the sense of a location. The following are most commonly used patterns for this purpose.

(1) location+comma+NP(headed by 'city')
e.g. Chicago, an old city
(2) 'city of' +location1+comma+location2
e.g. city of Albany, New York
(3) 'city of' +location
(4) 'state of'+location
(5) location1+{,}+location2+{,}+location3
e.g. (i) Williamsville, New York, USA
(ii) New York, Buffalo,USA
(6) {on, in}+location
e.g. on Strawberry → NeIsland
in Key West → NeCity

Patterns (1) , (3), (4) and (6) can be used to decide if the location is a city, a state or an island, while patterns (2) and (5) can be used to determine both the sub-tag and its sense. These patterns are implemented in our finite state transducer formalism.

## 4 Maximum Spanning Tree Calculation with Global Information

Although local context can be reliable evidence for disambiguating location senses, there are still many cases which cannot be captured by the above patterns. Information in the entire document (i.e. discourse information) should be considered. Since all location names in a document have meaning relationships among them, a way to represent the best sense combination within the document is needed.

The LocNZ process constructs a weighted graph where each node represents a location sense, and each edge represents similarity weight between location names. Apparently there will be no links among the different senses of a location name, so the graph will be partially complete. We calculate the maximum weight spanning tree (MaxST) using Kruskal's MinST algorithm (Cormen et al, 1990). The nodes on the resulting MaxST are the most promising senses of the location names.

We define three criteria for similarity weight assignment between two nodes:

(1) More weight will be given to the edge between a city and the province (or the country) to which it belongs.

(2) Distance between location names mentioned in the document is taken into consideration. The shorter the distance, the more we assign the weight between the nodes.

(3) The number of word occurrences affects the weight calculation. For multiple mentions of a location name, only one node will be represented in the graph. We assume that all the same location mentions have the same meaning in a document following *one sense per discourse* principle (Gale, Church, and Yarowsky, 1992).

When calculating the weight between two location names, the predefined similarity values shown in Table 1, the number of location name occurrences and the distance between them in a text are taken into consideration. After selecting each edge, the senses that are connected will be chosen, and other senses of the same location name will be discarded so that they will not be considered again in the MaxST calculation. A weight value is calculated with equation (1), where $s_{ij}$ indicate the $j^{th}$ sense of $word_i$, $a$ reflects the number of location name occurrences in a text, and $b$ refers to the distance between the two location names. Figure 4 shows the graph for calculating MaxST. Dots in a circle mean the number of senses of a location name.

Table 1. Similarity value $sim(s_i,s_i)$ between location sense pairs.

| Loc1 | Loc2 | Relationship | $Sim(s_i,s_i)$ |
|---|---|---|---|
| $C_1$ | $P_1$ | $P_1$ includes $C_1$ | 5 |
| $IL$ | $Ctr_1$ | $Ctr_1$ includes $IL$ | 5 |
| $C_1$ | $Ctr_1$ | $Ctr_1$ is direct parent | 5 |
| $C_1$ | $C_2$ | $C_1$ and $C_2$ in same province/state | 3 |
| $C_1$ | $C_2$ | $C_1$ and $C_2$ in same country | 2 |
| $C_1$ | $P_1$ | $C_1$ and $P_1$ are in same country but $C_1$ is not in $P_1$ | 2 |
| $C_1$ | $Ctr_1$ | *Ctr1 is not a direct parent of* $C_1$ | 3 |
| $P_1$ | $Ctr_1$ | $P_1$ is in $Ctr_1$ | 1 |
| $P_1$ | $P_2$ | $P_1$ and $P_2$ in same country | 1 |
| $Loc_1$ | $Loc_2$ | $Loc_1$ and $Loc_2$ are two sense nodes of the same location name | -∞ |
| $Loc_1$ | $Loc_2$ | Other cases | 0 |

Note: $C_i$: city; $P_i$: province/state; $IL$: island; $Ctr_i$: country; $Loc_i$: location.

$$Score(s_{ij}, s_{jk}) = sim(s_{ij}, s_{jk}) + a(s_{ij}, s_{jk}) - b(s_{ij}, s_{jk}) / numAll$$
$$a(s_{ij}, s_{jk}) = (num(w_i) + num(w_j)) / numAll$$
$$b(s_{ij}, s_{jk}) = dist(w_i, w_j) \quad (1)$$

## 5 Default Sense Extraction

In our experiments, we found that the system performance suffers greatly from the lack of lexical information on *default senses*. For example, people refer to "Los Angeles" as the city at California more than the city in Philippines, Chile, Puerto Rico, or the city in Texas in the USA. This problem becomes a bottleneck in the system performance. As mentioned before, a location name usually has a dozen senses that need sufficient evidence in a document for selecting one sense among them.

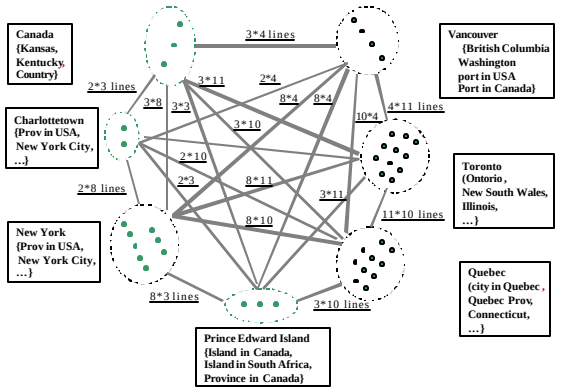


Figure 4. Graph for calculating maximum weight spanning tree.

But in many cases there is no explicit clue in a document, so the system has to choose the default senses that most people may refer to under common sense.

The Tipster Gazetteer (http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat) used in our system has 171,039 location entries with 237,916 total senses that cover most location names all over the world. Each location in the gazetteer may have several senses. Among them 30,711 location names have more than one sense. Although it has ranking tags on some location entries, a lot of them have no tags attached or the same rank is assigned to the entries of the same name. Manually calculating the default senses for over 30,000 location names will be difficult and it is subject to inconsistency due to the different knowledge background of the human taggers. To solve this problem in calculating the default senses of location names, we propose to extract the knowledge from a corpus using statistical processing method.

With the TREC-8 (Text Retrieval Conference) corpus, we can only extract default senses for 1687 location names, which cannot satisfy our requirement. This result shows that the general corpus is not sufficient to suit our purpose due to the serious 'data sparseness' problem. Through a series of experiments, we found that we could download highly useful information from Web search engines such as Google, Yahoo, and Northern Light by searching ambiguous location names in the Gazetteer. Web search engines can provide the closest content by their built-in ranking mechanisms. Among those engines, we found that the Yahoo search engine is the best one for our purpose. We wrote a script to download web-pages from Yahoo! using each ambiguous location name as a search string.

In order to derive default senses automatically from the downloaded web-pages, we use the similarity features and scoring values between location-sense pairs described in Section 3. For example, if "Los Angeles" co-occurs with "California" in the same web-page, then its sense will be most probably set to the city in California by the system. Suppose a location word $w$ has several city senses $s_i$: $Sense(w)$ indicates the default sense of $w$; $sim(w_i,x_{jk})$ means the similarity value between two senses of the word $w$ and the $j^{th}$ co-occuring word $x_j$; $num(w)$ is the number of $w$ in the document, and $NumAll$ is the total number of locations. $\alpha$ is a parameter that reflects the importance of the co-occurring location names and is determined empirically. The default sense of $w$ is $w_i$ that maximizes the similarity value with all co-occurring location names. The maximum similarity should be larger than a threshold to keep meaningful default senses. The threshold can be determined empirically through experimentation.

$$Sense(w) = \max_{1 \le i \le m} \sum_{j=1}^{n} \max_{1 \le k \le p} (sim(s_i, x_{jk}) * \alpha * (num(x_j)) / (NumAll - num(w))) \quad (2)$$

For each of 30,282 ambiguous location names, we used the name itself as search term in Yahoo to download its corresponding web-page. The system produced default senses for 18,446 location names. At the same time, it discarded the remaining location names because the corresponding web-pages do not contain sufficient evidence to reach the threshold. We observed that the results reflect the correct senses in most cases, and found that the discarded location names have low references in the search results of other Web search engines. This means they will not appear frequently in text, hence minimal impact on system performance. We manually modified some of the default sense results based on the ranking tags in the Tipster Gazetteer and some additional information on population of the locations in order to consolidate the default senses.

Table 2. Experimental evaluation for location name normalization.

| Document Type | No. of Ambigu-ous Loc Names | No. of Ambigu-ous senses | Correctly tagged locations | | | Precision (%) of LocNZ |
|---|---|---|---|---|---|---|
| | | | With Tipster Gazetteer default sense and rule only | With LocNZ default senses only | LocNZ | |
| California Intro. | 26 | 326 | 13 | 18 | 25 | 96 |
| Canada Intro. | 14 | 75 | 13 | 13 | 14 | 100 |
| Florida Intro | 22 | 221 | 10 | 18 | 20 | 90 |
| Texas Intro. | 13 | 153 | 9 | 11 | 12 | 93 |
| CNN News 1 | 27 | 486 | 10 | 23 | 25 | 92 |
| CNN News 2 | 26 | 360 | 10 | 22 | 24 | 92 |
| CNN News 3 | 16 | 113 | 4 | 10 | 14 | 87.5 |
| New York Times 1 | 8 | 140 | 1 | 7 | 8 | 100 |
| New York Times 2 | 10 | 119 | 2 | 7 | 10 | 100 |
| New York Times 3 | 18 | 218 | 5 | 13 | 17 | 94 |
| Total | 180 | 2211 | 77 (42%) | 142 (78%) | 169 (93.8%) | 93.8 |

## 6 Algorithm and Experiment

With the information from local context, discourse context and the knowledge of default senses, the location normalization process turned out to be very efficient and precise. The processing flow is divided into 5 steps:

Step 1. Look up the location gazetteer to associate candidate senses for each location NE;

Step 2. Call the pattern matching sub-module to resolve the ambiguity of the NEs involved in local patterns like "Williamsville, New York, USA" to retain only one sense for the NE as early as possible;

Step 3. Apply the 'one sense per discourse' principle for each disambiguated location name to *propagate* the selected sense to its other occurrences within a document;

Step 4. Call the global sub-module, which is a graph search algorithm, to resolve the remaining ambiguities;

Step 5. If the decision score for a location name is lower than a threshold, we choose a default sense of that name as a result.

For evaluating the system performance, 53 documents from a travel site (http://www.worldtravelguide.net/navigate/region/nam.asp), CNN News and New York Times are used. Table 2 shows some sample results from our test collections. For results shown in Column 4, we first applied default senses of location names available from the Tipster Gazetteer in accordance with the rules specified in the gazetteer document. If there is no ranking value tagged for a location name, we select the first sense in the gazetteer as its default. This experiment showed accuracy of 42%. For Column 5, we tagged the corpus with default senses we derived with the method described in section 5, and found that it can resolve 78% location name ambiguity. Column 6 in Table 2 is the result of our LocNZ system using the algorithm described above as well as default senses we derived. The system showed promising results with 93.8% accuracy.

## 7 Conclusion

This paper presents a method of location normalization for information extraction with experimental results and its applications. In future work, we will integrate a expanded location gazetteer including names of landmarks, mountains and lakes such as Holland Tunnel (in New York, not in Holland) and Hoover Dam (in Arizona, not in Alabama), to enlarge the system coverage, and adjust the scoring weight given in Table 1 for better normalization results. Using context information other than location names can be a subtask for determining specific location names such as bridge or area names.

## 8 Acknowledgement

The authors wish to thank Carrie Pine of AFRL for supporting this work. Other members of Cymfony's R&D team, including Sargur N. Srihari, have also contributed in various ways.

## References

Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest. 1990. *Introduction to Algorithm.* The MIT Press, pp. 504-505.

Dagon, Ido and Alon Itai. 1994. Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics*, Vol.20, pp. 563-596.

Gale, W.A., K.W. Church, and D. Yarowsky. 1992. One Sense Per Discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop.* pp. 233-237.

Hirst, Graeme. 1987. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge.

Krupka, G.R. and K. Hausman. 1998. IsoQuest Inc.: Description of the NetOwl (TM) Extractor System as Used for MUC-7. *Proceedings of MUC.*

McRoy, Susan W. 1992. Using Multiple Knowledge Sources for Word Sense Discrimination. *Computational Linguistics*, 18(1): 1-30.

Ng, Hwee Tou and Hian Beng Lee. 1996. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: an Exemplar-based Approach. In *Proceedings of 34th Annual Meeting of the Association for Computational Linguistics*, pp. 40-47, California.

Srihari, Rohini, Cheng Niu, and Wei Li. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. In *Proceedings of ANLP 2000*, Seattle.

Yarowsky, David. 1992. Word-sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of the 14th Internaional Conference on Computational Linguistics (COLING-92)*, pp. 454-460, Nates, France.

Yarowsky, David. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts.

## Appendix: Sample text and tagged result

Few countries in the world offer as many choices to the world traveler as Canada. Whether your passion is skiing, sailing, museum-combing or indulging in exceptional cuisine, Canada has it all.

Western Canada is renowned for its stunningly beautiful countryside. Stroll through Vancouver's Park, overlooking the blue waters of English Bay or ski the slopes of world-famous Whistler-Blackcomb, surrounded by thousands of hectares of pristine forestland. For a cultural experience, you can take an Aboriginal nature hike to learn about Canada's First Nations' history and cuisine, while outdoorsmen can river-raft, hike or heli-ski the thousands of kilometers of Canada's backcountry, where the memories of gold prospectors and pioneers still flourish today.

By contrast, Canada mixes the flavor and charm of Europe with the bustle of trendy New York. Toronto boasts an irresistible array of ethnic restaurants, bakeries and shops to tempt the palate, while Charlottetown, Canada's birthplace, is located amidst the rolling fields and sandy Atlantic beaches of Prince Edward Island. Between the two, ancient Quebec City is a world unto itself: the oldest standing citadel in North America and the heart of Quebec hospitality.

| Location | City | Province | Country |
|---|---|---|---|
| Canada | - | - | Canada |
| Vancouver | Vancouver | British Columbia | Canada |
| New York | New York | New York | USA |
| Toronto | Toronto | Ontario | Canada |
| Charlotte-town | Charlotte-town | Prince Edward Island | Canada |
| Prince Edward Island | - | Prince Edward Island | Canada |
| Quebec | Quebec | Quebec | Canada |