InfoXtract location normalization: a hybrid approach to geographic references in 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. Based on our previous work, this paper presents a refined hybrid approach to geographic references using our information extraction engine *InfoXtract*. The InfoXtract location normalization module consists of local pattern matching and discourse co-occurrence analysis as well as default senses. Multiple knowledge sources are used in a number of ways: (i) pattern matching driven by local context, (ii) maximum spanning tree search for discourse analysis, and (iii) applying default sense heuristics and extracting default senses from the web. The results are benchmarked with 96% accuracy on our test collections that consist of both news articles and tourist guides. The performance contribution for each component of the module is also benchmarked and discussed.

1 Introduction

The task of location normalization is to decode geographic references for extracted location Named Entities (NE). Ambiguity is a very serious problem for location NEs. For example, there are 23 cities named 'Buffalo', including the city in New York State and in the state of Alabama. Country names such as 'Canada', 'Brazil', and 'China' are also city names in the USA. Such ambiguity needs to be properly handled before converting location names into normal form to support Entity Profile (EP) construction, information merging/consolidation as well as visualization of *location-stamped* extracted events on a map.

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 an annotated corpus to perform WSD. Recent work emphasizes a 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.

In terms of local context, we found that there are certain fairly predictable keyword-driven patterns which can decide the senses of location NEs. These patterns use keywords such as 'city', 'town', 'province', 'on', 'in' or candidate location subtypes that can be assigned from a location gazetteer. For example, the pattern "X + city" can determine sense tags for cases like "New York City"; and the pattern "Candidate-city-name + comma + Candidate-state-name" 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 provide evidence for predicting the most probable sense of a location name. More specifically, location normalization depends on co-occurrence

* This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-01-C-0035. The authors wish to thank Carrie Pine of AFRL for supporting and commenting this work.

constraints of geographically related location entities mentioned in the same document. 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.

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.

Last but not least, proper assignment of default senses is found to play a significant role in the performance of a location normalizer. This involves two issues: (i) determining default senses using heuristics and/or other methods, such as statistical processing for semi-automatic default sense extraction from the web [Li *et al.* 2002]; and (ii) setting the conditions/thresholds and the proper levels when assigning default senses, to coordinate with local and discourse evidence for enhanced performance. The second issue can be resolved through experimentation.

In the light of the above overview, this paper presents an effective hybrid location normalization approach which consists of local pattern matching and discourse co-occurrence analysis as well as default senses. Multiple knowledge sources are used in a number of ways: (i) pattern matching driven by local context, (ii) maximum spanning tree search for discourse analysis, and (iii) applying heuristics-based default senses and web-extracted default senses in proper stages.

In the remaining text, Section 2 introduces the background for this research. Section 3 describes our previous work in this area and Section 4 presents the modified algorithm to address the issues with the previous method. Experiment and benchmarks are described in Section 5. Section 6 is the conclusion.

2 Background

The design and implementation of the location normalization module is an integrated part of Cymfony's core information extraction (IE) engine *InfoXtract*. InfoXtract extracts and normalizes entities, relationships and events from natural language text. Figure 1 shows the overall system architecture of InfoXtract, involving multiple modules in a pipeline structure.

InfoXtract involves a spectrum of linguistic processing and relationship/event extraction. This engine, in its current state, involves over 100 levels of processing and 12 major components. Some components are based on hand-crafted pattern matching rules, some are statistical models or procedures, and others are hybrid (e.g. NE, Co-reference, Location Normalization). The basic information extraction task is NE tagging [Krupka and Hausman 1998; Srihari *et al.* 2000]. The NE tagger identifies and classifies proper names of type PERSON, ORGANIZATION, PRODUCT, NAMED-EVENTS, LOCATION (LOC) as well as numerical expressions such as MEASUREMENT (e.g. MONEY, LENGTH, WEIGHT, etc) and time expressions (TIME, DATE, MONTH, etc.). Parallel to *location normalization*, InfoXtract also involves *time normalization* and *measurement normalization*.

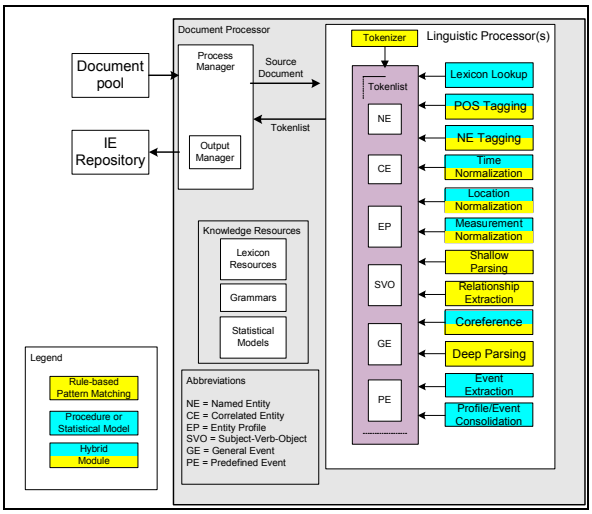

Figure 1: System Architecture of InfoXtract

InfoXtract combines the Maximum Entropy Model (MaxEnt) and Hidden Markov Model for NE tagging [Srihari *et al.* 2000]. Maximum Entropy Models incorporate local contextual evidence to handle ambiguity of information from a location gazetteer. In the Tipster Location Gazetteer used by InfoXtract, there are many 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. The NE tagger in InfoXtract only assigns the location super-type tag LOC to the identified location words and leaves the task of location sub-type tagging such as CITY or STATE and its disambiguation to the subsequent module Location Normalization.

Beyond NE, the major information objects extracted by InfoXtract are Correlated Entity (CE) relationships (e.g. AFFILIATION and POSITION), Entity Profile (EP) that is a collection of extracted entity-centric information, Subject-Verb-Object (SVO) which refers to dependency links between

logical subject/object and its verb governor, General Event (GE) on *who did what when and where* and Predefined Event (PE) such as *Management Succession* and *Company Acquisition*.

It is believed that these information objects capture the key content of the processed text. When normalized location, time and measurement NEs are associated with information objects (events, in particular) based on parsing, co-reference and/or discourse propagation, these events are *stamped.* The processing results are stored in *IE Repository*, a dynamic knowledge warehouse used to support cross-document consolidation, text mining for hidden patterns and IE applications. For example, location-stamped events can support information visualization on maps (Figure 2); time-stamped information objects can support visualization along a timeline; measurement-stamped objects will allow advanced retrieval such as *find all Company Acquisition events that involve money amount greater than 2 million US dollars.*

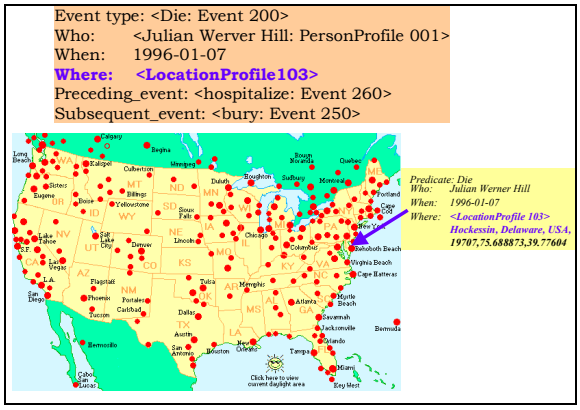

Figure 2: Location-stamped Information Visualization

3 Previous Work and Issues

This paper is follow-up research based on our previous work [Li *et al.* 2002]. Some efficiency and performance issues are identified and addressed by the modified approach.

The previous algorithm [Li *et al.* 2002] for location normalization consisted of five steps.

Step 1. Look up location names in the 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 [Gale *et al.*1992] for each disambiguated location name to *propagate* the selected sense to its other mentions within a document;

Step 4. Call the discourse sub-module, which is a graph search algorithm (*Kruskal*'s 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.

In this algorithm, Step 2, Step 4, and Step 5 complement each other, and help produce better overall performance.

Step 2 uses 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 the most commonly used patterns for this purpose.

(1) LOC + ',' + NP (headed by 'city')
e.g. Chicago, an old city
(2) 'city of' + LOC1 + ',' + LOC2
e.g. city of Albany, New York
(3) 'city of' + LOC
(4) 'state of' + LOC
(5) LOC1+ ',' + LOC2 + ',' + LOC3
e.g. (i) Williamsville, New York, USA
(ii) New York, Buffalo, USA
(6) 'on'/ 'in' + LOC
e.g. on Strawberry → ISLAND
in Key West → CITY

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.

Step 4 constructs a weighted graph where each node represents a location sense, and each edge represents similarity weight between location names. The graph is partially complete since there are no links among the different senses of a location name. The maximum weight spanning tree (MST) is calculated using Kruskal's MinST algorithm [Cormen *et al.* 1990]. The nodes on the resulting MST are the most promising senses of the location names.

Figure 3 and Figure 4 show the graphs for calculating MST. Dots in a circle mean the number of senses of a location name.

Through experiments, we found an efficiency problem in Step 4 which adopted Kruskal's algorithm for MST search to capture the impact of location co-occurrence in a discourse. While this

algorithm works fairly well for short documents (e.g. most news articles), there is a serious time complexity issue when numerous location names are contained in long documents. A weighted graph is constructed by linking sense nodes for each location with the sense nodes for other locations. In addition, there is also an associated performance issue: the value weighting for the calculated edges using the previous method is not distinctive enough. We observe that the number of location mentions and the distance between the location names impact the selection of location senses, but the previous method could not reflect these factors in distinguishing the weights of candidate senses.

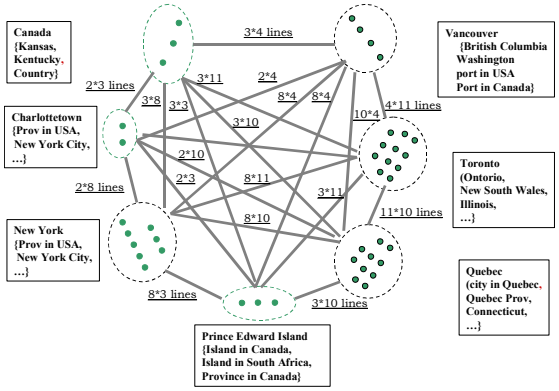

Figure 3: Graph and its Spanning Tree

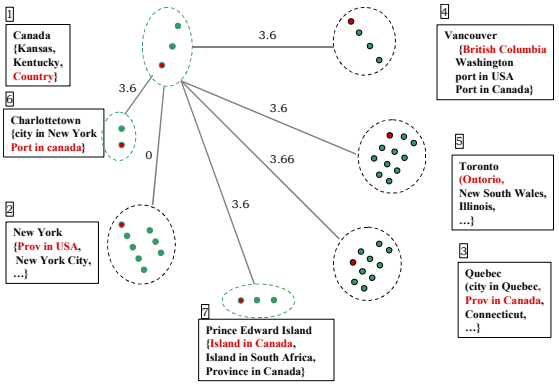

Figure 4: Max Spanning Tree

Finally, our research shows that default senses play a significant role in location normalization. For example, people refer to "Los Angeles" as the city in *California* more than the city in the *Philippines, Chile, Puerto Rico,* or the city in *Texas* in the *USA*. Unfortunately, the available Tipster Gazetteer (http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat) does not mark default senses for most entries. It has 171,039 location entries with 237,916 senses, among which 30,711 location names are ambiguous. Manually tagging the default senses for over 30,000 location names is difficult; moreover, it is also subject to inconsistency due to the different knowledge backgrounds of the human taggers. This problem was solved by developing a procedure to automatically extract default senses from web pages using the *Yahoo!* search engine [Li *et al.* 2002]. Such a procedure has the advantage of enabling 're-training' of default senses when necessary. If the web pages obtained through *Yahoo!* represent a typical North American 'view' of what default sense should be assigned to location names, it may be desirable to re-train the default senses of location names using other views (e.g. an Asian view or African view) when the system needs to handle overseas documents that contain many foreign location names.

In addition to the above automatic default sense extraction, we later found that a few simple default sense heuristics, when used at proper levels, can further enhance performance. This finding is incorporated in our modified approach described in Section 3 below.

4 Modified Hybrid Approach

To address the issues identified in Section 2, we adopt *Prim*'s algorithm, which traverses each node of a graph to choose the most promising senses. This algorithm has much less search space and shows the advantage of being able to reflect the number of location mentions and their distances in a document.

The following is the description of our adapted *Prim*'s algorithm for the weight calculation.

The weight of each sense of a node is calculated by considering the effect of linked senses of other location nodes based on a predefined weight table (Table 1) for the sense categories of co-occurring location names. For example, when a location name with a potential city sense co-occurs with a location name with a potential state/province sense and the city is in the state/province, the impact weight of the state/province name on the city name is fairly high, with the weight set to 3 as shown in the 3rd row of Table 1.

Table 1. Impact weight of Sense2 on Sense1

Sense1	Sense2	Condition	Weight
City	*City*	in same state	2
	City	in same country	1
	State	in same state	3
	Country	in country without state (e.g. in Europe)	4

Let $W(S_i)$ be the calculated weight of a sense S_j of a location; $weight(S_j \rightarrow S_i)$ means the weight of S_i influenced by sense S_j; $Num(Loc_i)$ is the number of location mentions; and $\beta/dist(Loc_i, Loc_j)$ is the

measure of distance between two locations. The final sense of a location is the one that has maximum weight. A location name may be mentioned a number of times in a document. For each location name, we only count the location mention that has the maximum sense weight summation in equation (1) and eventually propagate the selected sense of this location mention to all its other mentions based on *one sense per discourse* principle. Equation (2) refers to the sense with the maximum weight for Loc_i.

$$(1)\quad W(S_i) = \sum_{j=0}^{m} weight(S_j \rightarrow S_i) * Num(Loc_j) * \left(\beta / dist(Loc_i, Loc_j)\right)$$

$$(2)\quad S(Loc_i) = \arg\max_j (W(S_j))$$
$$0 \le j \le w$$

Through experiments, we also found that it is beneficial to select default senses when candidate location senses in the discourse analysis turn out to be of the same weight. We included two kinds of default senses: heuristics-based default senses and the default senses extracted semi-automatically from the web using Yahoo. For the first category of default senses, we observe that if a name has a country sense and other senses, such as "China" and "Canada", the country senses are dominant in most cases. The situation is the same for a name with province sense and for a name with country capital sense (e.g. London, Beijing). The updated algorithm for location normalization is as follows.

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

Step 2. If a location has sense of country, then select that sense as the default sense of that location (heuristics);

Step 3. Call the pattern matching sub-module for local patterns like "Williamsville, New York, USA";

Step 4. Apply the 'one sense per discourse' principle for each disambiguated location name to *propagate* the selected sense to its other mentions within a document;

Step 5. Apply default sense heuristics for a location with province or capital senses;

Step 6. Call Prim's algorithm in the discourse sub-module to resolve the remaining ambiguities (Figure 5);

Step 7. If the difference between the sense with the maximum weight and the sense with next largest weight is equal to or lower than a threshold, choose the default sense of that name from lexicon. Otherwise, choose the sense with the maximum weight as output.

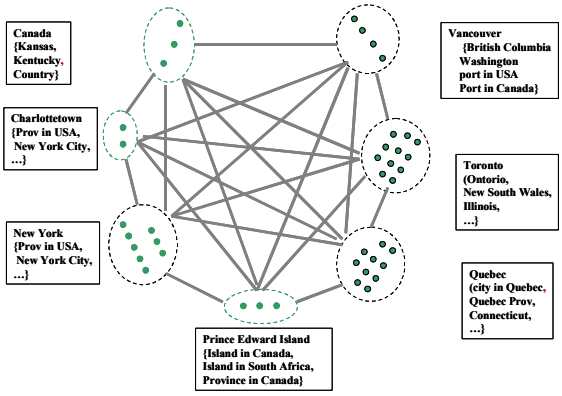

Figure 5: Weight assigned to Sense Nodes

5 Experiment and Benchmark

With the information from local context, discourse context and the knowledge of default senses, the location normalization process is efficient and precise.

The testing documents were randomly selected from CNN news and from travel guide web pages. Table 2 shows the preliminary testing results using different configurations.

As shown, local patterns (Column 4) alone contribute 12% to the overall performance while proper use of defaults senses and the heuristics (Column 5) can achieve close to 90%. In terms of discourse co-occurrence evidence, the new method using *Prim's* algorithm (Column 7) is clearly better than the previous method using *Kruskal*'s algorithm (Column 6), with 13% enhancement (from 73.8% to 86.6%). But both methods cannot outperform default senses. Finally, when using all three types of evidence, the new hybrid method presented in this paper shows significant performance enhancement (96% in Column 9) over the previous method (81.9% in Column 8), in addition to a satisfactory solution to the efficiency problem.

Table 2. Experimental evaluation for location normalization

File	# of ambiguous location names	# of mentions	Pattern hits	Def-senses	Kruskal Algo. only	Prim Algo only	Kruskal +Pattern +Def (previous)	Prim +Pattern +Def (new)
Cnn1	26	39	4	20	21	24	26	26
Cnn2	12	20	5	11	7	10	11	11
Cnn3	14	29	0	12	10	12	10	14
Cnn4	8	14	2	8	4	4	4	8
Cnn5	11	26	1	9	5	8	5	9
Cnn6	19	35	6	16	11	16	13	18
Cnn7	11	27	0	11	4	7	6	10
Calif.	16	30	0	16	16	16	16	16
Florida	19	28	0	19	19	19	18	19
Texas	13	13	0	12	13	13	13	12
Total	149	261	12%	89.9%	73.8%	86.6%	**81.9%**	**96%**

We observed that if a file contains more concentrated locations, such as the state introductions in the travel guides for California, Florida and Texas, the accuracy is higher than the relatively short news articles from CNN.

6 Conclusion and Future Work

This paper presented an effective hybrid method of location normalization for information extraction with promising experimental results. In the future, we will integrate an 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. Meanwhile, more extensive benchmarking is currently being planned in order to conduct a detailed analysis of different evidence sources and their interaction and contribution to system performance.

References

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

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

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

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

Huifeng Li, Rohini K. Srihari, Cheng Niu, Wei Li. 2002. Location Normalization for Information Extraction, *COLING 2002*, Taipei, Taiwan.

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. *ACL 1996*, 40-47, California.

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

Yarowsky, David. 1992. Word-sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *COLING 1992*, 454-460, Nantes, France.

Yarowsky, David. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *ACL 1995*, Cambridge, Massachusetts.