Place disambiguation with co-occurrence models.

Simon Overell[1], João Magalhães[1] and Stefan Rüger[2,1]
[1]Multimedia & Information Systems
Department of Computing, Imperial College London, SW7 2AZ, UK
[2]Knowledge Media institute
The Open University, Milton Keynes, MK7 6BJ, UK
{simon.overell, j.magalhaes} @imperial.ac.uk and s.rueger@open.ac.uk

Abstract

In this paper we describe the geographic information retrieval system developed by the Multimedia & Information Systems team for GeoCLEF 2006 and the results achieved. We detail our methods for generating and applying co-occurrence models for the purpose of place name disambiguation, our use of named entity recognition tools and text indexing applications. The presented system is split into two stages: a batch text & geographic indexer and a real time query engine. The query engine takes manually crafted queries where the text component is separated from the geographic component. Two monolingual runs were submitted for the GeoCLEF evaluation, the first constructed from the title and description, the second included the narrative also.

We explain in detail our use of co-occurrence models for place name disambiguation using a model generated from Wikipedia.

The paper concludes with a full description of future work and ways in which the system could be optimised.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software;

General Terms

Measurement, Performance, Experimentation

Keywords

Geographic Information Retrieval, Disambiguation, Wikipedia, Co-occurrence

1 Introduction

In this paper we detail the MMIS team's entry for GeoCLEF 2006. We have two objectives with our entry: to test the accuracy of our co-occurrence model generated from Wikipedia and to test whether the use of large scale co-occurrence models can aid the disambiguation of geographic entities.

We begin with a discussion of disambiguation methods, followed by a full outline of the system, we then present our experimental runs and results, concluding with an analysis and future work.

Methods of place name disambiguation can generally be split into three categories:

- *Rule-Based methods*, which use a series of hand crafted heuristic rules [4, 5, 10, 12, 15, 19, 21].

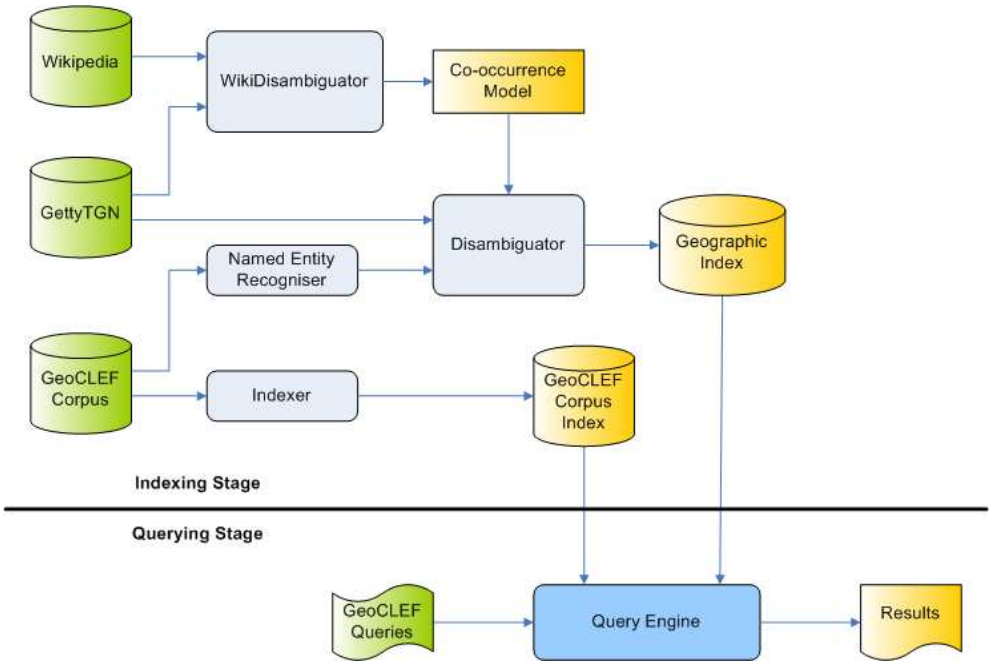


Figure 1: System Design

- *Data Driven methods*, which require a large annotated corpus that standard data mining rules can be applied to [6, 7].
- *Semi-Supervised methods*, which require a smaller annotated corpus (but multiple examples of each ambiguity) and an additional un-annotated corpus [2, 11, 13, 16].

We use a *rule-based* approach to annotate how places occur in Wikipedia (taking advantage of structure and meta-data). This annotated corpus is then applied as a co-occurrence model using a *data-driven* method to annotate the GeoCLEF data.

2 The system

Our geographic information retrieval system is split into two parts: the *indexing stage* and the *querying stage*. The Indexing stage requires the corpus and some external resources to generate the geographic and text indexes (a *slow* task). The querying stage requires the generated indexes and the queries; it runs in *real time*.

The Indexing stage consists of four separate applications: *WikiDisambiguator* is first used to build a co-occurrence model of how place names occur together in Wikipedia [14]; *Disambiguator* then applies the co-occurrence model to disambiguate the named entities extracted from the GeoCLEF corpus with *Named Entity Recogniser*. The disambiguated named entities form the geographic index; *Indexer* is used to build the text index.

The Querying stage consists of our *Query Engine*, which queries the text index and geographic index separately, combining the results (Figure 1).

2.1 WikiDisambiguator

Wikipedia is being used more and more in geographic information retrieval, it is extremely useful as a resource due to its size, variation, accuracy and quantity of hyper-links and meta-data. Anyone can contribute articles to Wikipedia meaning the diversity of articles is huge: to date there are

over 2 million articles and stubs (short articles) [18]. In GIR it has been used for corpus [14], ontology [3], gazetteer [3, 4, 14] and ground truth [14] generation. The places extracted from Wikipedia are correlated with the Getty Thesaurus of Geographic Names (TGN), a gazetteer listing approximately 800,000 places.

WikiDisambiguator is the application designed to build our co-occurrence model. The data gathered (collected from a crawl of every Wikipedia article[1]) takes the form of three database tables: links believed to be places and the order in which they occur; links believed to be non-places and the order in which they occur and a mapping of Wikipedia articles to TGN unique identifiers.

WikiDisambiguator uses rule-based methods of disambiguation. It is made up of two parts, the disambiguation framework and the method of disambiguation itself. Using Wikipedia as the corpus solves two problems: the problem of *synonyms* (multiple words referring to a single entity) is resolved as we can record how multiple anchor texts point to the same page; and the problem of *polynyms* (a single word referring to multiple places) can be solved with our disambiguation system.

2.1.1 The disambiguation framework

The disambiguation framework is a simple framework to allow independent disambiguation methods to be slotted in.

The framework is outlined as follows:

```
1: The Wikipedia articles are loaded from the database
2: for each Wikipedia article all the links are extracted
3:   for each link
4:     if it has already been disambiguated as not a place
5:       then add an entry to the db and continue
6:     if the page pointed to has already been disambiguated as a place
7:       then add an entry to the db and continue
8:     else attempt to disambiguate using the Method of Disambiguation specified
9:   end for
10: end for
```

The disambiguation methods is passed:

- a list of candidate places
- a list of names of places related to this link
- the text making up the article that this link points to
- the article title
- how the link appeared in the text

The candidate places are taken from the Getty Thesaurus of Geographical Names. The candidate places for an article are places matching either the article's title or the anchor text linking to the article.

2.1.2 Our method of disambiguation

Based on the results observed by running a series of simple disambiguation methods on test data, we designed a disambiguation pipeline that could exploit the meta-data contained in Wikipedia and strike a balance between precision and recall [14].

Each disambiguation method is called in turn (Figure 2). A list of candidate places is maintained for each article, an article is denoted as unambiguous when this list contains one or zero elements. Each method of disambiguation can act on the candidate places list in the following

[1]Our copy of Wikipedia was taken 3rd Dec 2005

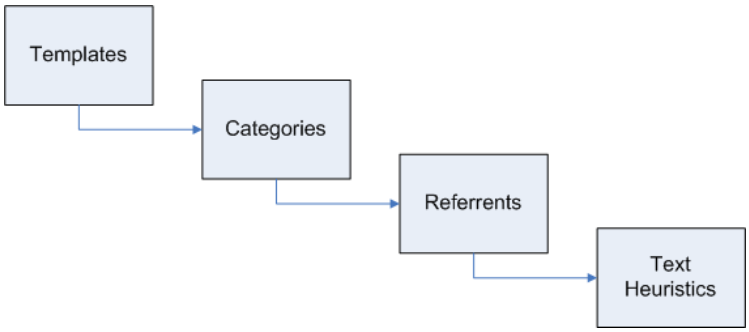


Figure 2: Disambiguation Pipeline

ways: remove a candidate place; add a candidate place; remove all candidate places (disambiguate as not a location); or remove all bar one candidate places (disambiguate as a location).

1. *Disambiguate with templates* – The template data in Wikipedia is highly formatted data contained in name-value pairs. The format of the templates is as follows {{*template name* | *name* = *value* | ...}}. The template name is used initially for disambiguation, for example "Country" will indicate this page refers to a location of feature type nation or country. Templates are also used to identify non-places, for example if the template type is "Biographic" or "Taxonomic." The name-value pairs within a template are also used for disambiguation, e.g. in the Coord template a latitude and longitude are provided which can be matched to the gazetteer.

2. *Disambiguate with categories* – The category information from Wikipedia contains softer information than the template information [9]; the purpose of assigning documents to categories is to denote associations between documents (rather than template information which is intended to display information in a uniform manner). Category tags can identify the country or continent of an article or indicate an article is not referring to a place.

3. *Disambiguate with referents* – Often in articles describing a place, a parent place will be mentioned (e.g. when describing a town, mention the county or country). The first paragraph of the document is searched for containing places. This method of disambiguation has been shown to have a suitably high percentage or precision (places correctly identified) and grounding (places correctly matched to unique identifiers) of 87% and 95% respectively [14].

4. *Disambiguate with Text Heuristics* – Our heuristic method is based on the hypothesis *When describing an important place*[2]*, only places of equal or greater importance are used as referrers.* This hypothesis is implemented as follows:

   1: All the place names are extracted from the first paragraph of the document
   2: **for each** possible location of the ambiguous place
   3: Sum the distance between the possible location and the extracted locations that are more important than this one.
   4: **end for**
   5: **return** the place with the minimal sum

2.2 Named Entity Recogniser

News articles have a large number of references to named entities that quickly place the reader into the context of the news piece. Sometimes the same named entity is referred to in different ways

[2]In our implementation importance is based on the feature type recorded in the gazetteer.

(e.g. "British prime minister", "Mr. Blair", "Tony Blair"). Thus, the detection of references to all named entities is the problem that we addressed in this part of the system. This part receives as input the GeoCLEF news articles and outputs the named entities of each news article, which will be used by the *Disambiguator*.

Named entity recognition systems rely on lexicons and textual patterns either manually crafted or learnt from a training set of documents. We used the ESpotter named entity recognition system proposed by Zhu et al. [20]. Currently, ESpotter recognises people, organisations, locations, research areas, email addresses, telephone numbers, postal codes, and other proper names. ESpotter has the particularity of supporting domains of interest. First it infers the domain of the document (e.g. computer science, sports, politics) to adapt the lexicon and patterns for a more specialised named entity recognition which will result in a high precision and recall.

ESpotter uses a database to store the lexicon and the textual pattern information. It can be easily customised to recognise any type of entities one might be interested in by adding new lexicon and textual patterns. The database we used is the one supplied by Zhu et al., we did not create a database of GeoCLEF based lexicon and patterns.

2.3 News articles indexing

The news article corpus was indexed with Apache Lucene 2.0 [1], which was later used to search the article corpus. The information retrieval model we used was the vector space model without term frequencies (binary term weight). This decision was due to the small size of each document that could cause a large bias for some terms. Terms are extracted from the news corpus in the following way:

1. Split words at punctuation characters, removing punctuation; however, a dot that's not followed by whitespace is considered part of a term;
2. Split words at hyphens and generate a term: unless there is a number in the term, in which case the whole term is interpreted as a product number and is not split
3. Recognise email addresses and internet host names as one term
4. Remove every stop word
5. Index a document by its extract terms (lowercase)

See [1] for details.

2.4 Disambiguator

To allow the returned results to be pruned geographically the data needs to be geographically indexed. We take the named entities tagged as locations output by the *Named Entity Recogniser* and disambiguate them based on how they co-occur in our co-occurrence model.

Having the corpus indexed with place names, we could apply our co-occurrence model to disambiguate the places to distinct locations as listed in the Getty TGN. Our method is a Naïve Bayesian approach designed to maximise speed and implemented in the application **Disambiguator**.

```
1: for all documents
2:   for each adjacent tuple of place names
3:     for each possible location for either place name
4:       disambiguate as the places that most often appear together
5:     end for
6:   end for
7: end for
```

Possible locations are defined as any location appearing in our co-occurrence model that has been referred to by the same toponym as the named entity extracted from the corpus. The geographic index is then stored in a Postgres database and indexed with an R-Tree (to allow efficient processing of spatial queries) [8, 17]. In previous experiments we have shown the co-occurrence model to be accurate to within 80% [14], in this experiment we assume the geographic index to have an accuracy equal to or less than this.

2.5 Query Engine

The *Query Engine* is the application used to prune the results of the text queries produced by Lucene using the geographic queries.

The queries are manually split into a text component and a geographic component. The text query is handled normally by Lucene, the geographic query is manually split into a tree of conjunctions and disjunctions.

2.5.1 Executing a text query

Once the news articles are indexed with Lucene, the query terms will be extracted in the same way that the document terms were, a similarity measure is taken between the query's terms and all indexed documents. The similarity function is given by the following expression:

$$\text{score}(q,d) = \frac{\sum_{t \in d} \text{tf}_t(d) \cdot \text{idf}^2(d \ni t, D) \cdot \text{norm}(d)}{\sqrt{\sum_{t \in d} \text{tf}_t^2(d)}},$$

where $\sum_{t \in d} \text{tf}_t(d)$ is the t term frequency for the given document d (in our case is 0 or 1), $\text{idf}(d \ni t, D)$ is the frequency of documents d containing the term t in the D collection, and $\text{norm}(d)$ is a normalization constant given by the total number of terms in document d. See [1] for details.

2.5.2 The query tree

The query trees are constructed by hand. The nodes of the tree are either conjunctions or disjunctions while the leaves of the tree are (spatial-relation, location) pairs see Figure 3.

2.5.3 Executing a query

For each document that matches the text query we check whether it refers to a place matching the geographic query – any documents not matching the geographic query are removed (Figure 4).

1: **fetch all** documents that satisfy the text query from Lucene
2: **for all** documents
3: prune results against the geographic query
4: **end for**
5: **return** remaining documents

3 Experimental runs

We entered two runs for GeoCLEF 2006: Both were mono-lingual, English queries on an English corpus with manually constructed queries. Our first run used queries constructed from the title and description, the second run also took into account the narrative. As far as was possible we attempted to add no world knowledge, the query trees we produced resembled what could be produced with a query parser.

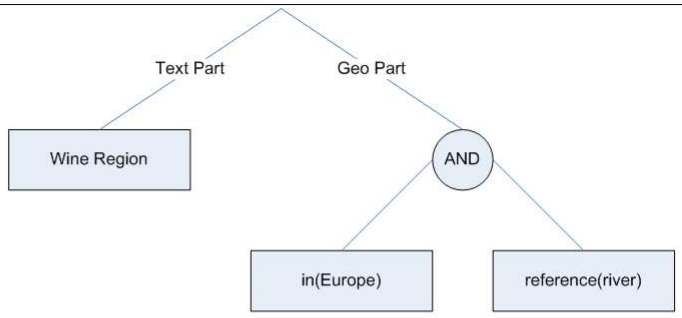
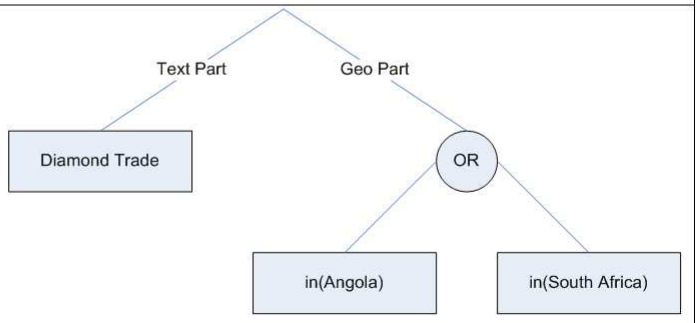
Query	Tree
Wine regions around rivers in Europe	
Diamond trade in Angola and South Africa	

Figure 3: Query Trees

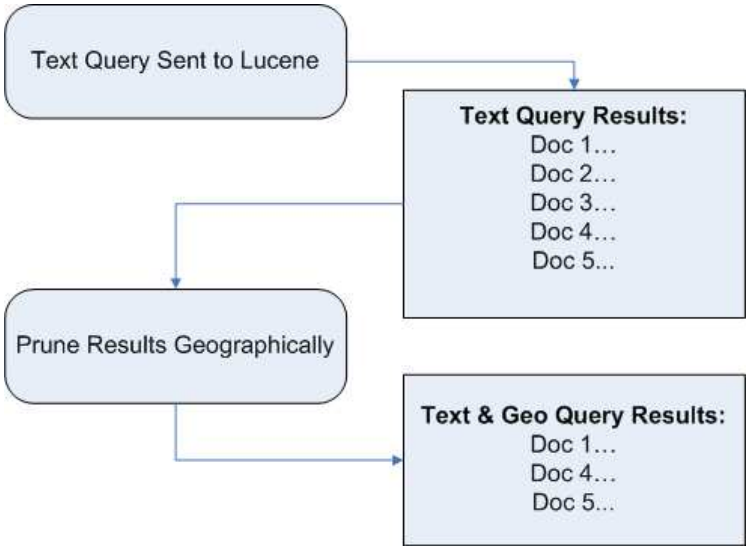


Figure 4: Executing a query

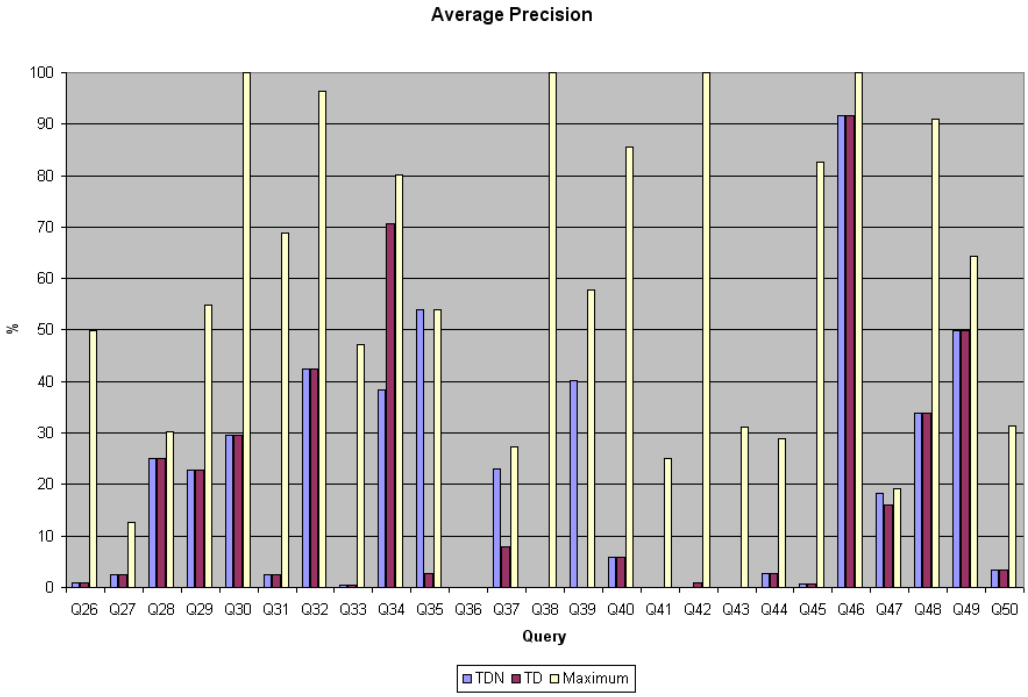


Figure 5: Comparison of Average Precision

4 Results

Our runs appeared between the 25% quantile and the median for mean average precision (see table below). The run consisting of queries constructed from Title, Description and Narrative (TDN) generally out performed the run constructed from Title and Description (TD); in Figure 5 we compare the average precision of our runs for each query against the maximum average precision achieved by any system.

Mean Average Precision						
TDN	TD	Worst	Q1	Median	Q3	Best
19.53%	16.49%	4%	15.64%	21.62%	24.59%	32.23%

5 Conclusions

Our system as presented here uses a simple approach to the application of co-occurrence models for place name disambiguation, text indexing and the combination of text and geographic queries.

The system gave results appearing slightly below the median MAP; this shows the system model is valid; however, there is significant room for improvement. Without further tests we cannot comment on specific parts of the system; each of the five applications will have to be tuned independently.

With respect to our objectives we can conclude that the co-occurrence model accuracy agrees with the previous experiments conducted in [14] and that co-occurrence models are a suitable method of place name disambiguation.

6 Future Work

We are currently exploring whether we can improve our results by applying co-occurrence models in more sophisticated ways. The three methods currently being worked on are:

- Using a generalised Jaccard co-efficient to produce a co-occurrence index
- Learning a hierarchical decision list
- Applying Latent Semantic Indexing to build place-name neighbourhoods

We hope after a study of these methods to evaluate the suitability of using co-occurrence models for place-name evaluation and to identify the optimal method.

ESpotter utilises an Access database for Named Entity Recognition; we would like to see if it is possible to optimise this database for use with the GeoCLEF corpus and general place name recognition.

Lucene was applied in the default configuration and the text part of the queries were not altered in any way. We plan to experiment with suitable query weights for Lucene and try alternative configurations of the index. Ultimately we would like to combine the geographic and text indexes so that they can be searched and applied simultaneously.

We also plan to implement a query parser to allow the queries to automatically be parsed into query trees; this would require a level of natural language processing.

References

[1] Apache Lucene Project. http://lucene.apache.org/java/docs/, 18/Aug/2006.

[2] B. Bucher, P. Clough, D. Finch, H. Joho, R. Purves, and A. Syed. Evaluation of SPIRIT prototype following integration and testing. Technical report, 2005.

[3] D. Buscaldi, P. Rosso, and P. P. Garcia. Inferring geographic ontologies from multiple resources for geographic information retrieval. In *SIGIR Workshop on Geographic Information Retrieval*, pages 52–55, 2006.

[4] N. Cardoso, B. Martins, M. S. Chaves, L. Andrade, and M. J. Silva. The XLDB group at GeoCLEF 2005. In *Working Notes for the GeoCLEF 2005 Workshop*, 2005.

[5] P. Clough, M. Sanderson, and H. Joho. Extraction of semantic annotations from textual web pages. Technical report, 2004.

[6] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. In *Journal of the Society for Information Science*, volume 41, pages 391–407, 1990.

[7] D. A. Grossman and O. Frieder. *Information Retrieval*. Springer-Verlag, second edition, 2004.

[8] A. Guttman. R-Trees, A dynamic index structure for spatial searching. In *Proceedings of SIGMOD*, pages 47–57. ACM Press, 1984.

[9] D. Kinzler. Wikisense - Mining the Wiki. In *Proceedings of Wikimania 05*, 2005.

[10] J. L. Leidner, G. Sinclair, and B. Webber. Grounding spatial named entities for information extraction and question answering. 2003.

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

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

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

[14] S. Overell and S. Rüger. Identifying and grounding descriptions of places. In *SIGIR Workshop on Geographic Information Retrieval*, pages 14–16, 2006.

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

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

[17] The PostgreSQL Global Development Group. *PostgreSQL 8.1.2 Documentation*, 2005.

[18] Wikipedia. http://www.wikipedia.org, 18/Aug/2006.

[19] A. G. Woodruff. Gipsy: Georeferenced information processing system. Technical report, 1994.

[20] J. Zhu, V. Uren, and E. Motta. ESpotter: Adaptive named entity recognition for web browsing. In *Proc. of Professional Knowledge Management Conference*, pages 518–529. Springer-Verlag, 2005.

[21] W. Zong, D. Wu, A. Sun, E. Lim, and D. H. Goh. On assigning place names to geography related web pages. In *Proceedings of JCDL*, pages 354–362, 2005.