MMIS at GeoCLEF 2008: Experiments in GIR

Simon Overell[1], Adam Rae[2] 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 6AA, UK
seo01@doc.ic.ac.uk and {a.rae, s.rueger}@open.ac.uk

Abstract

In this paper we present our Geographic Information Retrieval System, Forostar, and the results of three experiments. We compare two data fusion methods, and show that a simple geographic filter outperforms a penalty based system. We compare context based disambiguation to a default gazetteer and show no significant difference. Finally we compare a unique geographic index to an ambiguous geographic index. The ambiguous index outperformed all other methods and was statistically significantly better than the baseline.

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; H.3.7 Digital Libraries

General Terms

Measurement, Performance, Experimentation

Keywords

Geographic Information Retrieval, Placename Disambiguation, Geographic Indexing, Geographic Relevance Ranking

1 Introduction

This paper presents the work from the MultiMedia Information Systems group at GeoCLEF 2008 with our GIR application Forostar. We submitted nine runs performing three experiments.

Our first experiment compares two data fusion methods looking at how best to combine text and geographic relevance data. Our text results take the form of a rank returned from a standard IR system, while our geographic results take the form of a filter – an unranked list of all documents matching the geographic part of the query. We compare penalising documents not appearing in the filter with a learned penalisation value to simply applying the filter to the text rank.

Our second experiment compares context based placename disambiguation to a default gazetteer. We have a unique geographic index mapping each placename reference in a document to a single location on the Earth's surface. This mapping is not a trivial matter for ambiguous locations, such as "Cambridge" or "London". We compare three ways of generating this index, the simplest matches each placename to the most common location with that name. We compare this to two methods that use other placenames occurring in the document to provide a context to

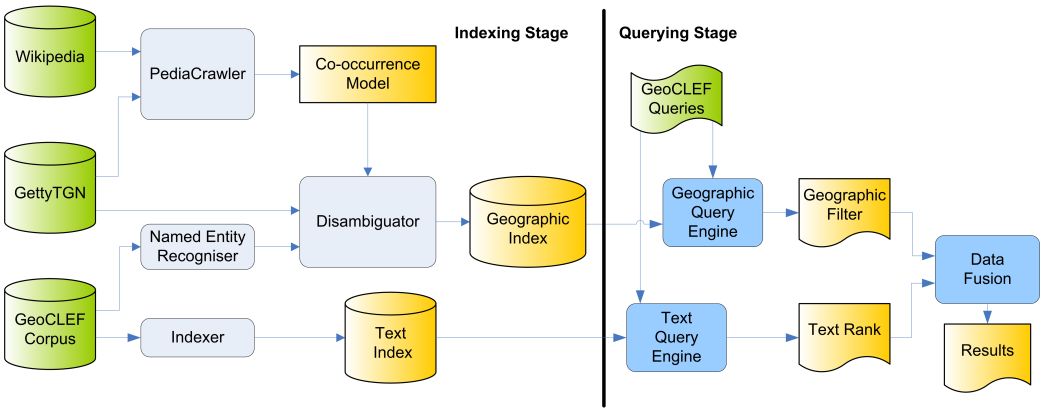

Figure 1: Forostar framework

disambiguate the ambiguous placenames. A model mined from Wikipedia is used as training data.

Our final experiment compares our unique index to an ambiguous index, in an ambiguous index each ambiguous placename is indexed multiple times, once for each possible location. An ambiguous index will clearly provide significantly greater recall but at a potential cost to precision (dependent on the accuracy of the unique index).

In Section 2 we outline our GIR system and the different methods used in our experiments. Section 3 will reiterate our research questions and describe our submitted runs. Section 4 contains our results, significance testing and comparisons to the other GeoCLEF submissions. We conclude with Section 5 containing our observations and planned future work.

2 System

Forostar is our ad-hoc Geographic Information Retrieval system (Figure 1). At indexing time, documents are analysed and named entities extracted. Named entities tagged as locations are then disambiguated using our co-occurrence model extracted from Wikipedia [8]. The free-text fields are then indexed by Lucene. In the querying stage we query the Geographic and Text indexes separately combining them in the data-fusion module.

2.1 Indexing Stage

Forostar's **Indexer** is based on Apache Lucene [9]. Text fields are pre-processed by a customised analyser similar to Lucene's default analyser: Text is split at white space into tokens, the tokens are then converted to lower case, stop words discarded and stemmed with the "Snowball Stemmer". The processed tokens are held in Lucene's inverted index.

The **Named Entity Reconiser** used to process the text fields is Sheffield University's General Architecture for Text Engineering (GATE) [1]. The bundled Information Extraction Engine, ANNIE, performs named entity recognition, extracting named entities and tagging them as locations. Our disambiguation system matches these placenames to unique locations in the Getty Thesaurus of Geographical Names (TGN) [3].

PediaCrawler, the application that builds our geographic co-occurrence model, is not discussed in detail in this paper; instead we refer the reader to [8]. Suffice to say the co-occurrence model produced acts as training data for our placename disambiguation algorithms described below. The co-occurrence model used is available by contacting the lead author.

We compare four alternative modules for the **Disambiguator** described below. All map the placenames that occur in a document to locations in the TGN and store them in a geographic

Cambridge, UK	Preston	Derby	Sheffield	Lincoln	King's Lynn
Cambridge, MA	Barnstaple	Bristol	Dukes	Essex	Middlesex
Cambridge, NZ	Wuxi	Alexandra	Ashburton	Carterton	Coromandel

Table 1: Example trigger words for the placename 'Cambridge'

index.

- **Most Referred to (MR).** The most referred to method matches each each placename to the most referred to location with that placename in the co-occurrence model. Essentially we build a default gazetteer providing a many-to-one mapping of placenames to locations.
- **Support Vector Machine (SVM).** We construct a multi-dimensional vector space, which is partitioned with an SVM [5]: ambiguous placenames are the classification objects, possible locations the classification classes, co-occurring locations are the features, and the features have $tf\bullet idf$ weights. We train a separately classifier for each possible location–placename mapping and classify as the location whose classifier outputs the greatest decision value. The co-occurrence model forms our training data.
- **Neighbourhoods (Neigh).** For each possible location of an ambiguous placename we build a collection of trigger words. Trigger words are other placenames, which if they appear in the same document as the ambiguous placename identify the location. If none of the trigger words appear the placename is classified using the MR method.

  Trigger words are found based on a relatedness score. [2] describe the relatedness score as the ratio of co-occurrences between words divided by the disjunction of occurrences. This can be defined as:

  $$r(x, y) = \frac{f_{xy}}{f_x + f_y - f_{xy}}, \tag{1}$$

  where f_x denotes the total number of times word x appears. The top 5 trigger words for three different locations for the placename 'Cambridge' are shown in Table 1.
- **No Disambiguation (NoDis).** Although not strictly speaking a method of disambiguation, the NoDis method provides an alternative to a disambiguation algorithm. NoDis builds an ambiguous index, indexing each ambiguous placename once for each possible location. For example the placename 'Cambridge' would be indexed 3 times: Cambridge, UK, Cambridge MA, and Cambridge NZ.

2.2 Querying Stage

The text and geographic indexes are queried separately by two different query engines. Placenames are manually extracted from queries and passed to the Geographic Query Engine[1]. Queries are passed to the text query engine with no pre-processing.

We use the standard Lucene **Text Query Engine**. This performs a comparison between the documents and the query in a $tf\bullet idf$ weighted vector space. The cosine distance is taken between the query vector and the document vectors. The text rank is produced in standard TREC format.

We also use Lucene to store the **Geographic Index**. A *unique string* is formed for each location, this is the TGN id of this location, preceded with the TGN id of all the parent locations, separated with slashes. Thus the unique string for the location "London, UK" is the TGN id for London (7011781), preceded by its parent, Greater London (7008136), preceded by its parent, Britain (7002445)... until the root location, the World (1000000) is reached. Giving the unique string for London as 1000000\1000003\7008591\7002445\7008136\7011781.

[1]The only reason this was not handled by the Named Entity Reconiser was time constraints in the implementation.

Disambig. Method	Penalisation Values
MR	2.0
SVN	3.0
Neigh.	8.0
NoDis.	3.0

Table 2: Penalisation values found by using results from GeoCLEF 2005 – 2007 as training data.

The **Geographic Query Engine** disambiguates each placename passed to it using the MR method of disambiguation[2]. The locations are then converted into unique strings and Lucene's *ConstantScoreRangeQuery* is used to search the index fast and build a geographic filter (also in TREC format).

Storing and searching locations in this fashion means all locations contained within a larger location are also included in a query. So a query for "United States" will produce a filter including documents mentioning all the states, counties and towns within the United States as well as references to the country itself.

The **Data Fusion** module combines the text rank and the geographic filter to produce a single result rank. Two different Data Fusion methods were used described below:

- **Penalisation**. The penalisation method multiplies the rank, r, of each element in the text rank that is not in the geographic filter by a penalisation value p, to give an intermediate rank $r\prime$. This can be described by the filter function $f(r)$

$$f(r) = \begin{cases} r & \text{doc } r \text{ present in filter} \\ rp & \text{doc } r \text{ not present in filter} \end{cases} \quad (2)$$

  The intermediate rank is sorted by $r\prime$ to give the final returned rank.

  The penalisation value p is found using a brute force search using the 75 queries and relevance judgements from GeoCLEF 2005 – 2007 as training data. The search finds the value of p that maximises mean average precision (MAP). The p values found for each disambiguation method are shown in Table 2.

- **Filtering**. The filtering method reorders the text rank in a more aggressive way than the penalisation method. All the results of the text rank that are also contained in the geographic filter are returned first, followed by the text results that are not in the geographic filter. This is equivalent to the penalisation method with a p value of ∞.

An example of these two methods are shown in Figure 2. It shows a hypothetical text rank containing two entries also in the geographic filter. The penalisation method calculates $r\prime$, shown in brackets, to re-order the results, while the filter method simply promotes all the documents also in the geographic filter to the top of the rank.

3 Experiments

Our experiments are all monolingual. We use the 25 English Language GeoCLEF queries and the English Language corpus consisting of $\approx 170,000$ documents from the Glasgow Herald and Los Angeles Times. GeoCLEF topics contain title, description and narrative fields. We only use the title field as this field bears most similarity to the types of geographic query submitted to search engines.

As explained in Section 1 we are performing three experiments at this year's GeoCLEF. This involved us submitting nine runs: a simple text baseline (the text rank before data fusion) and

[2]This method is chosen due to the minimal context contained in queries.

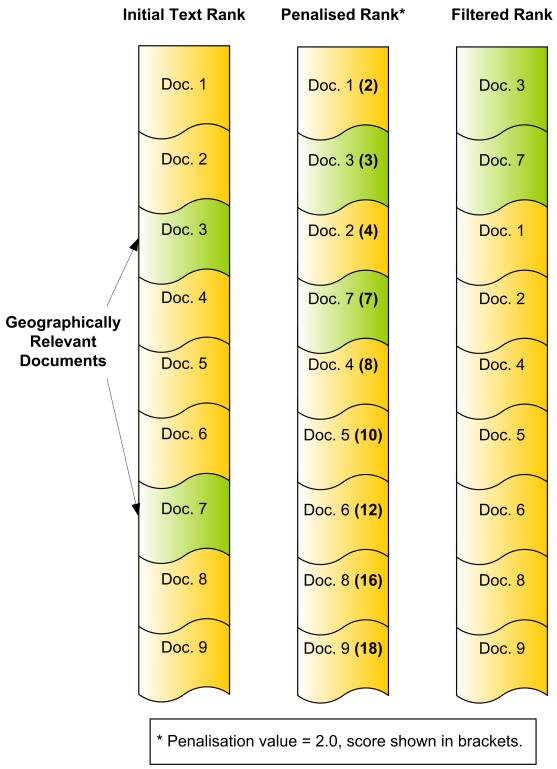

Figure 2: Rank Example

both of the combination methods combined with each of the four disambiguation methods. Our experimental questions are outlined below in brief:

- Can the more sophisticated Penalisation data fusion method outperform the more aggressive Filtering method?
- Can the more sophisticated context based disambiguation methods Neigh and SVM outperform the default gazetteer MR method?
- Are the unique geographic index methods accurate enough to outperform the NoDis method?

4 Results

The results of our nine runs are displayed in Table 3. Mean Average Precision (MAP) is the metric primarily used in GeoCLEF, however Geometric Average Precision (Geo AP) is also shown to give an indication of how consistent the methods are. Notice that combining the geographic information using the penalisation filter actually gives us worse results than the text baseline. Our assumption here is that the penalisation training on the past GeoCLEF data is over fitting. On the other hand, the filter method outperforms the baseline in every case showing it to be more robust.

There is minimal difference between which disambiguation method is used regardless of the fusion method.

We performed pairwise statistical significance testing of each method with the baseline using the Wilcoxon signed rank test rejecting the null hypothesis only when $p < 5\%$ [4]. We found that all the Penalisation results were statistically significantly worse than the baseline, and only the NoDis-Filter method was statistically significantly better. We also performed pairwise significance

Disambig.	Fusion	MAP	Geo AP
Text Baseline		24.1%	6.52%
MR	Penalis.	18.9%	5.12%
SVN	Penalis.	18.9%	5.17%
Neigh.	Penalis.	19.0%	5.24%
NoDis.	Penalis.	18.9%	5.17%
MR	Filter	24.5%	11.22%
SVN	Filter	24.4%	7.85%
Neigh.	Filter	24.2%	7.88%
NoDis.	Filter	26.4%	10.98%

Table 3: GeoCLEF 2008 results

Quartile	MAP
Best	30.4%
Q3	26.1%
Median	23.7%
Q1	21.4%
Worst	16.1%

Table 4: GeoCLEF 2008 quartiles

testing between the Penalisation method and Filter method runs with the same disambiguation method and found that in every case the Filter method was statistically significantly better.

4.1 Comparison with other participants

For comparison with the rest of the participants at GeoCLEF 2008, Table 4 shows the best, worst and quartile ranges of all the submitted runs. Our best result, NoDis-Filter, occurs in the top quartile. The other filtered results and the baseline occur between the Median and Q3. The Penalisation results occur in the lower quartile.

5 Conclusions

In past years at GeoCLEF many submitted methods augmented with Geographic Information (including our past attempts) have not provided a statistically significant improvement over a baseline [7, 6]. We were pleased to buck this trend this year with our NoDis-Filter method, which re-orders the text result using an ambiguous geographic filter. The NoDis-Filter method was statistically significantly better than our text only baseline, it also came in the top quartile of methods submitted.

In response to our research questions it is still unclear whether an ambiguous or unique text index provides the best results, also whether context based disambiguation can improve over non-context based methods. We have, however, shown that using a penalisation value to combine text and geographic data is highly sensitive to over fitting and a simple filter much more robust. In fact using a brute force search to optimise the penalisation value resulted in an MAP on the test data statistically significantly worse than the baseline or filter methods.

5.1 Future work

In future work we would like to expand our context based disambiguation algorithms to take into account not only the content of the documents but also the associated meta-data, i.e. where the documents were published. We believe treating placenames in the Los Angeles Times differently

from the Glasgow Herald could produce further improvements in disambiguation and retrieval performance.

Also we would like to further develop the Penalisation data fusion method as it is potentially more powerful than the Filter method. Alternative methods of training the system will need to be explored, specifically how to avoid over fitting.

References

[1] H Cunningham, D Maynard, V Tablan, C Ursu, and K Bontcheva. Developing language processing components with GATE. Technical report, University of Sheffield, 2001.

[2] J A Guthrie, L Guthrie, Y Wilks, and H Aidinejad. Subject-dependent co-occurrence and word sense disambiguation. In *Meeting of the Association for Computational Linguistics*, pages 146–152, 1991.

[3] P Harping. *User's Guide to the TGN Data Releases*. The Getty Vocabulary Program, 2.0 edition, 2000.

[4] D Hull. Using statistical testing in the evaluation of retrieval experiments. In *SIGIR conference on Research and development in information retrieval*, pages 329–338, 1993.

[5] T Joachims. *Advances in Kernel Methods – Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT-Press, 1999.

[6] S Overell, J Magalhães, and S Rüger. Forostar: A system for GIR. In *Lecture Notes from the Cross Language Evaluation Forum 2006*, 2007.

[7] S Overell, J Magalhães, and S Rüger. GIR experiments with Forostar at geoCLEF 2007. In A Nardi and C Peters, editors, *CLEF 2007 Workshop, Working notes*, September 2007.

[8] S Overell and S Rüger. Geographic co-occurrence as a tool for GIR. In *CIKM Workshop on Geographic Information Retrieval*, 2007.

[9] Apache Lucene Project. http://lucene.apache.org/java/docs/. Accessed 1 August 2007, 2007.