UAIC: Participation in LAGI Task

Adrian Iftene

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
adiftene@info.uaic.ro

Abstract. The LogCLEF track was launched in 2009 with aim to analyse and classify the user's queries and had two tasks LAGI (Log Analysis and Geographic Query Identification) and LADS (Log Analysis for Digital Societies). In this edition from 2009, we built a system in order to participate in the LAGI task. The system uses GATE or Wikipedia like external resources in order to identify geographical entities in user's queries. Because, the results obtained using these resource are comparable, the main advantage of using GATE resources comes from the short duration of execution in comparison with using of Wikipedia resources. A brief description of our system is given in this paper.

1 Introduction

LogCLEF[1] deals with the analysis of queries as expression of user behavior. The goal is the analysis and classification of queries in order to improve search systems. LogCLEF had two tasks:

- *Log Analysis and Geographic Query Identification* (LAGI): The recognition of the geographic component within a query stream is a key problem for geographic information retrieval (GIR).
- *Log Analysis for Digital Societies* (LADS): This task used logs from The European Library (TEL) and had intention to analyze user behavior with a focus on multilingual search.

Our group sends runs only for LAGI task. The aim of the LAGI task is to identify geographic elements in search log queries. For that competitors' received two sets of logs:

1. Tumba! - a Portuguese web search engine
2. The European Library (TEL) - on line search for materials in various libraries in Europe.

The way in which we built the system for LAGI track is presented in Section 2, while Section 3 presents the runs submitted. Last Section presents conclusions regarding our participation in LAGI 2009.

[1] LogCLEF: http://www.uni-hildesheim.de/logclef/

2 UAIC System for LAGI

Our system searches only in the subset in English logs (which are the majority of the logs). Organizers force participating groups to respect some rules. The rules respected by our system were:

1. A query is a geographical query if and only if it is bounded geographically.
2. A place term can be any country, a city or town, mountain, province or region from GATE (Cunningham et al., 2001) or if it is described as a place in Wikipedia (Portuguese Wikipedia for Tumba! and English Wikipedia for our English subset of TEL).
3. A candidate place term can map to more than one possible meaning in Wikipedia.
4. A place term can occur in a title (of a book, movie, team, etc.), but the title itself (if a different text span from the place) is not to be tagged.
5. Capitalization (upper and lower case) in the query is ignored, as it is used inconsistently in the queries.
6. Wildcards ('*') are ignored.
7. If some words of a query can be interpreted as forming a phrase, this will be preferred over interpreting those words as isolated words put in the same query.

In order to participate in LAGI, additional to using Wikipedia resources offered by organizers, we built another resource with geographical name entities starting from GATE resources. This resource was loaded by our program in cache and it is used after that in identification of geographical resources. The Figure 1 presents the system architecture.

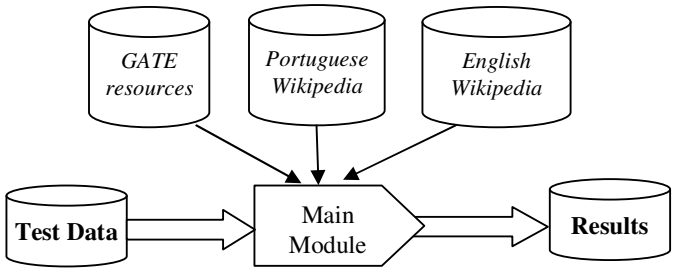

Figure 1: UAIC system used in LAGI

The used resources and the main module are presented below.

2.1 Test Data

The test data consists from two files, one from Tumba! with 152 entries and one from TEL with 108 entries. In comparison the training files had 7 entries for Tumba! file

and 9 entries for TEL file. The format for input data and for requested output data are presented below in Table 1 and in Table 2 and are taken from TEL training data.

Table 1: TEL input data for LAGI task

```
875336 & 5431 & ("central europe")
828587 & 12840 & ("sicilia")
902980 & 482 &  (creator all "casanova")
196270 & 5365 & ("casanova")
528968 & 190 &  ("iceland*")
470448 & 8435 & ("iceland")
712725 & 5409 & ("cavan county ireland 1870")
```

Table 2: TEL output data for LAGI task

```
875336 & 5431 & ("<place>central europe</place>")
828587 & 12840 & ("<place>sicilia</place>")
902980 & 482 &  (creator all "casanova")
196270 & 5365 & ("casanova")
528968 & 190 &  ("<place>iceland</place>*")
470448 & 8435 & ("<place>iceland</place>")
712725 & 5409 & ("<place>cavan county ireland</place> 1870")
```

How we can see in above tables the aim is to add <place> </place> tags to user queries for geographical elements.

2.2 Resources

In order to build our resource with geographical entities we start from GATE resources and additional, we search on the web in order to add new similar entities. In separated runs, we load our resources or resources provide by organizers: page titles from Portuguese and English Wikipedia.

GATE

From GATE (Cunningham et al., 2001) we use the following sets of named entities: *cities*, *countries*, *small regions*, *regions*, *mountains* and *provinces*. In the end the total number of entities used from GATE resources was 146.581 and the size on disk was around 4.47 Mb. In Table 3 are examples of GATE resources.

Table 3: Examples of GATE resources

Type	Examples	Entities Number
City	Aaccra, Aalborg, Aarhus, Ababa, Abadan, Abakan, Aberdeen, Abha, …	143.487
Country	Afghanistan, Afrique, Albania, Albanie, Alderney, Algeria, …	465
Mountain	Alps, Andes, Himalaya, Pyrenees, Snowdonia	5
Province	Aana, Aargau, Abaco, Abruzzi, Abyan,	2.411

	Aceh, Acores, Acquaviva, …	
Region	Africa, Algarve, Antarctica, Ashmore and Cartier Islands, Asia, …	213

Wikipedia

Organizers offered to the participants' two files with titles from Portuguese and English Wikipedia. In comparison with GATE, the number of entries from file with Portuguese titles from Wikipedia was 934.395 and the size on disk was 35.3 Mb, and the number of entries from file with English titles from Wikipedia was 6.996.744 and the size on disk was 282 Mb. In Table 4 are few lines from these files.

Table 4: Examples from Wikipedia Resources

Language	Examples
English	<title>AmericanSamoa</title> <title>AppliedEthics</title> <title>AccessibleComputing</title> <title>Anarchism</title>…
Portuguese	<title>Astronomia</title> <title>Astronomia e astrofísica</title> <title>América Latina</title> <title>Albino Forjaz de Sampaio</title>…

2.3 Main Module

The main module load successively the GATE or Wikipedia resources in cache using a hash map in which the key is named entity itself, and the value is the number of words from initial named entity. After that using this cache we will try to identify in TEL and Tumba test data the geographical entities.

Resources Loading

From beginning when we load our geographical resources in cache, we transform all characters from these entities in lower case. Additional, we split every name entity in components words and load also these separated words in our cache, but we specify the number of words from initial entity (in this way we will know if this key from our hash map come from a simple name entity or from a composed name entity). We will see how we will use this value when we try to identify geographical entities in user query. Even if we lose much time at resource loading, after that the operations of identification are very fast regardless of the number of user's queries that we want to process.

Test Data Pre-Processing

In order to identify in test data geographical entities from our cache, we perform pre-processing of test data. The most important steps are:

1) In first step we parse the current line from test data and extract only the relevant text (actually this is the initial user query).

2) Second step has the aim to ignore special characters like +, (,), *, “, ” or white spaces (tab, space, return) in initial user query.
3) Third step transform all characters extracted in previous step in lower case characters. The result is a new form of user query, called from now *new query*.

Geographical Entities Identification

The most important operation of main module is identification of the geographical entities in this new form of user query. From now the main question is: *How we identify the geographical entities in this new query*?

Initial we try to see if we have in our cache the new query itself. If YES, then we finished the process of identification of geographical entities and we skip to the next line in test data file. This is the case of the following line from Tel test data:

```
4752 & 11759 & ("portugal)"
```

for which we have in our cache the all user query which is “Portugal” from GATE countries file.

If NO, then we try to split new user query in separated words if this is possible. When we have only one word in the new query we automatically skip to the next line in test data file. When we have more than one word, we apply the following steps:

1. **At step 1** every individual word is searched in hash map. If current word comes from a simple named entity we simply add “place” tags to it. This is the case of below line:

   ```
   4892 & 5670 & ("climbing on the Himalaya and other mountain ranges)"
   ```

   for which we have in our cache the separated word “Himalaya” from GATE mountains file.

2. **At step 2** for every word searched in hash that comes from a composed named entity we look successively in it left and it right in order to combine more words with the same value in hash map.

   2.1. If we have like neighbors these types of hash keys then we create a common tag. This is the case of the following line:

   ```
   13128 & 11516 & ("peter woods)"
   ```

   for which we have in our cache separated keys “Peter” and “Woods” from GATE cities file (first one from “Peter Tavy” and second one from “Harper Woods”). Because both have the same value in hash (2 which represents the number of words from initial entity) we create a common tag for both words.

   2.2. If we haven’t like neighbors these types of hash keys, then we eliminate the all these tags.

During all previous steps the stop words are ignored.

3 Submitted Runs

We submitted two pairs of runs: one in which the main module loaded GATE like external resource, and one in which Portuguese or English Wikipedia are loaded like external resources.

Table 5: UAIC Runs

Test Data	Resource	Rcount	Hcount	Match	Prec	Rec	Fmeas
TEL	GATE	21	29	7	24.14	33.33	28.00
TEL	Wikipedia	21	99	16	16.16	76.19	26.67
Tumba	GATE	35	69	18	26.09	51.43	34.62
Tumba	Wikipedia	35	147	13	8.84	37.14	14.29

Differences between results obtained with GATE and results obtained with Wikipedia are the following:

- With GATE we mark like geographical entities a lower number in comparison with Wikipedia (29 in comparison with 99 for TEL test data, and 69 in comparison with 147 for Tumba test data) while the correct marches are comparable and this is the reason for higher precisions in GATE case (24.14% in comparison with 16.16% for TEL test data, and 26.09% in comparison with 8.84% for Tumba test data);
- Regarding correct matches, in case of TEL test data, Wikipedia offered more correct matches and in this case the recall is higher (76.19% in comparison with 33.33%). Amazingly, in case of Tumba test data the number of correct matches is higher again for GATE case (18 instead of 13) and obvious the recall is higher also in this case (51.43% in comparison with 37.14%). Because in this case precision and recall are both higher, in the end we have the greater F-measure: 34.62%.

4 Conclusions

This paper presents the UAIC system which took part in the LogCLEF 2009 competition in LAGI task. The system uses like external resources GATE files and two files offered by organizers with titles from Portuguese and English Wikipedia.

Initial we load external resources in cache using a hash map. After that, at pre-processing part we obtain a new query from the initial user query. This new query is used by main module in order to identify geographical entities. In this process we searches in cache the initial query, and after that words components, with aim to have in the end the most comprehensive succession of geographical entities.

The results show how the results obtained with GATE or with Wikipedia like external resources are comparable like quality.

Acknowledgements

The author would like to thank to the students Victor Chircu and Alexandru Cristea and their colleagues from group 4A, second year, for their help and support at different stages of system development.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. *In ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 168--175, Association for Computational Linguistics, Morristown, NJ, USA (2001)