UAIC: Participation in INFILE@CLEF task

Cristian-Alexandru Drăguşanu, Alecsandru Grigoriu, Andreea-Loredana Andreşan, Daniela Epure, Dan Anton, Adrian Iftene

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
{cristian.dragusanu, alecsandru.grigoriu, andreea.andresan, daniela.epure, dananton, adiftene}@info.uaic.ro

Abstract. This year marked UAIC[1]'s first participation at the INFILE@CLEF competition. This campaign's purpose is the evaluation of cross-language adaptive filtering systems, which is to successfully build an automated system that separates relevant from non-relevant documents written in different languages in an incoming stream of textual information with respect to a given profile. A brief description of our system, including presentation of the Parsing, Indexing and Filtering modules is given in this paper, as well as the results of the submitted runs.

1 Introduction

INFILE@CLEF[2] (information filtering evaluation) extends the TREC 2002 filtering track. In comparison, it uses a corpus of 100,000 Agence France Press comparable newswires for Arabic, English and French (Besançon et al., 2008). Also, the evaluation is performed using an automatic querying of test systems with a simulated user feedback. Each system can use the feedback at any time to increase performance. Test systems will provide Boolean decisions for each document and filter profile. INFILE was also open to monolingual participation. Coordinators were CEA (FR), U. Lille (FR), ELDA (FR).

The participants received news collections contains 100,000 news articles for each language (English, French and Arabic), stored in directories, and each news article is in a separate file, in XML format (NewsML format), encoded in UTF-8. Every tag in the XML may be used for filtering. The articles in the different languages are not translations of one another, they are independent articles. Also, the participants received 50 topics for all three languages.

In the batch filtering task, competitors must compare each topic in a source language to the documents in the target languages. Every source/target languages are allowed: results can be provided for monolingual filtering, cross lingual filtering or multilingual filtering (with a mixed set of documents from different target languages), as long as are used only the topics in the source language (provided translations of the topics should not be used for cross lingual filtering, either directly or for training).

[1] "Al. I. Cuza" University
[2] INFILE@CLEF: http://www.infile.org/

Any external resources can be used for cross lingual filtering (bilingual dictionaries, aligned corpora, machine translation, web etc). In the end, for each document, systems must assess the relevance of the document to the considered topic.

The way in which we built the system for INFILE track is presented in Section 2, while Section 3 is concerned with presentation of details related to evaluation of our system. Last Section presents conclusions regarding our participation in INFILE 2009.

2. UAIC System

Our system has three main modules: module one responsible with XML parsing, module two that indexes the XML files, and the third module that does the filtering. The Figure 1 presents the system architecture.

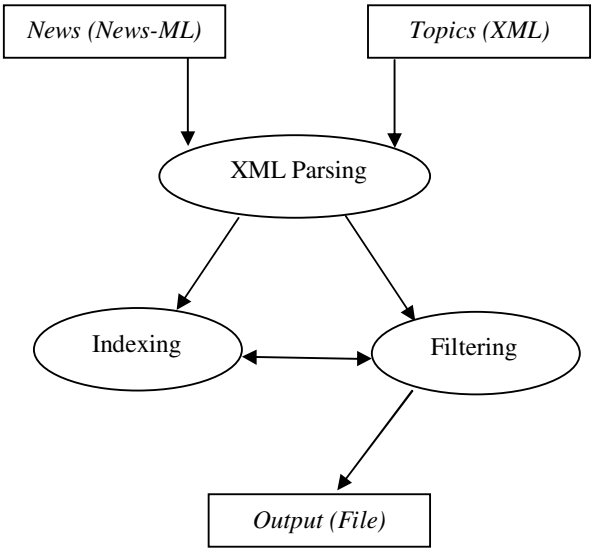

Figure 1: UAIC system used in INFILE 2009

Module details are presented below.

2.1 Module for XML Parsing

First of all, we parse the XML files with aim to parse and to extract relevant content from documents (which are in News-ML format).

The *Indexing module* needs several data form the NewsML documents, essential for the Filtering run. We extract that data using a parsing technique based on the XML Document Object Model (DOM) Parsing methods (Web Consortium) – the full

procedure will be presented in detail during this chapter. From the given files we need to focus on *DateID*, *NewsItemId*, *Slugline*, *Headline*, *DataContent*, *City*, *Country* and *Filename* (all important to the indexing part). The Topics store a minimum amount of data therefore we will focus only on: Topic number, Title, Description, Narration, Keywords and Sample (used later on by the Filtering module).

2.1.1 Parsing the News

In order to parse the news, we need a *global counter*, in order to remember how many NewsML documents have been already returned, to know which one to return next.

A *file path* is also needed for us to know where the list of NewsML documents is stored.

The main parsing procedure that parses the news needs to know two essential parameters: the *directory* where the NewsML documents will be stored and *file path* presented above. Based, on these parameters the method will retrieve and after that will return the desired data.

After we finish with one NewsML file, we process the next NewsML file from the list stored locally in the *input.txt* file with the desired data.

In all steps presented above, the XML files can be seen as trees; therefore the data can be represented as nodes. In the end the method will return the content of the node regardless to the child nodes.

2.1.2 Parsing the Topics

The method that parses the topics takes as parameter the absolute path to the XML that contains the topics. Another method returns the desired topic index for which we will later use on the Filtering part, obtaining the results based on that indexed topic.

We insert all the topics from the XML topic file (located with the xmlpath parameter) into an ArrayList and return the ArrayList with the desired data. Similar to the NewsML Parsing we work with trees and nodes, extracting and adding the information from the nodes.

2.2 Indexing Module

For indexing we use Lucene (Hatcher, E. and Gospodnetic, O., 2005) a suite of free libraries used both for indexing and for searching.

All documents are parsed by XML Parsing, one by one. From each document, a number of representative fields are stored (*DateID*, *NewsItemId*, *Slugline*, *Headline*, *DataContent*, *City*, *Country* and *Filename*) and sent to the Indexing module as parameters.

The Indexing module receives the relevant tags from each XML file and analyzes and stores that information in the main index database.

2.4 Filtering Module

The Filtering part can be viewed as a separate application, even though the used modules are the same as in the Indexing part.

In the Filtering part, the file containing the 50 topics (in XML format) is parsed by XML Parsing module. Then, for each one of the 50 topics, a number of fields are stored (*Topic number*, *Title*, *Description*, *Narration*, *Keywords* and *Sample*) and are sent to the Filtering module.

The Filtering module receives the topic details, sorts and filters individual words from all fields and generates a search query based on the most frequent relevant words from the topic. The search query is designed to optimize the index search by adding specific terms to be searched in specific index fields (for example *Slugline*, *Headline*, etc.) and by adding different importance to each field. When the query string is fully generated, it's passed as a parameter on to the Indexing module, which will return a list of documents matching the query.

For instance, let's consider the first topic for English, which looks like this:

```
<top>
  <num>101</num>
  <title>Fight against doping in sport</title>
  <desc>This profile includes information on the fight against
doping in sport, anti-doping legislation, banned substances, and
prevention.</desc>
  <narr>Relevant documents include information on the problem of
doping. They provide clear and official information on products
that may contain prohibiteb substances to athletes, the risks on
the health of individuals and penalties for use of doping
products.</narr>
  <keywords>
  <keyword>doping</keyword>
  <keyword>Legislation doping</keyword>
  <keyword>athletes</keyword>
  <keyword>doping substances</keyword>
  <keyword>Fight against doping </keyword>
  </keywords>
  <sample>On October 19, 2005, WADA welcomed with great
satisfaction the unanimous adoption of the first International
Convention against Doping in Sport by the General Conference of
UNESCO, at its plenary session."The adoption of the Convention by
UNESCO is a strong signal of the commitment of the governments of
the world to the fight against doping in sport," said David Howman,
WADA's Director General. "The drafting of this Convention in just
two years was a world record for international treaties. We warmly
commend and thank UNESCO for facilitating the process, and we look
forward to the treaty coming into force and the ratification by
each government."
  </sample>
</top>
```

First of all, the topic is parsed by the Parsing module, splitting it in multiple fields, like: TopicNumber, Title, Description, Narration, Keywords and Sample.

Then, the algorithm removes all prepositions from all fields, so these words won't be searched, being too general to return any relevant result.

After that, the most frequent 5 words and the negated expressions (preceded by words such as: *anti, except, not,* etc.) are extracted from all fields.

Also, a heuristic algorithm was implemented to try to extract dates or locations (cities and/or countries) from the fields. Because the News-ML format can contain information about date of the article, or the location, finding them in the topic might return better search results.

After all these items (frequent words, dates, locations) are extracted, they are combined in different ways, so a general search query can be formed and used to search the previously created index for matching documents.

The final search query looks like this (for the topic example given above):

```
(+"2012" +"olympic" +"games" +"organization") (DateId:[6/15/2003
TO 8/15/2003] OR DateId:[12/01/1999 TO 12/31/1999])
(HeadLine:"which"^7 OR HeadLine:"are"^7 OR HeadLine:"cities"^7
OR HeadLine:"olympic"^7)(Slugline: "which"^4 OR Slugline:
"are"^4 OR Slugline: "cities"^4 OR Slugline: "olympic"^4 OR
Slugline: "games"^4 OR Slugline: "organization"^4 OR Slugline:
"international"^4 OR Slugline: "committee"^4)(City: havana^3 OR
City: paris^3 OR City: madrid^3 OR City: rio^3 OR City: ny^3 OR
City: la^3 OR City: moscow^3 OR City: new york^3)("not be" OR
"which" OR "find" OR "documents" OR "expose" OR "selection" OR
"are" OR "applications" OR "several" OR "cities" OR "paris" OR
"submitted" OR "their" OR "2012" OR "nine" OR "olympic" OR
"games" OR "organization" OR "international" OR "committee")
```

First 3 words are the most frequent words in the topic title. Because of this, they are marked as the most important words in the query, and they are mandatory in all search fields (they are marked with a leading "+").

The second section contains detected dates, which will be searched in the *DateId* search field.

The next query section represents the words that will be searched in the *Headline* field. These words include the most frequent word from the *Description*, *Narration, Sample* and *Keywords* topic fields.

After that, there's a section containing the words which will be searched in the *Slugline* field. These include the most frequent word from *Description, Narration* and *Sample*, and all 5 frequent words from *Keywords*.

Also, because the topic contains some city names, they were added in a separate section. They will be searched in the *City* search field.

Finally, a section containing all frequent words from all topic fields, which will be searched in the *DataContent* search field.

The above example also includes field priorities, marked as a trailing "^" and a number. Field priorities set different score for the returned results, based on what words were found (and in what fields). Our results, on the other hand, were best with basic priority for all fields (so all fields had the same importance in determining the results).

The results will be written in the results file and the next topic is processed. If there are no more topics to be processed, the application stops.

2.5 Submitted Runs

For our runs the search was made in 2 languages, English and French, using topics in English. Each language archive contains 100,000 news articles, stored in directories according to the following organization:

```
<language>/<year>/<month>/<day>
```

Each news article is in a separate file, in XML format (NewsML format), encoded in UTF-8.

There are 50 topics for each language, but only the English topics were used for testing. The 50 topics are stored in one single file, encoded in UTF-8, in XML having the following format:

```
<topics>
  <top> topic node
  <num>...</num> topic identifier (numeric)
  <title>...</title> a short title
  <desc>...</desc> a descriptive sentence of the subject
  <narr>...</narr> a longer description of what is a relevant
  document
  <keywords>
       <keyword>...</keyword>
       <keyword>...</keyword>
       ...
  </keywords> a set of key words or key phrases (5 at most)
  <sample>...</sample> a sample excerpt of relevant document
  for the topic (not taken from the news collection to be
  filtered)
  </top>
  ...
<topics>
```

Usually, on a search on all fields, using the most optimized algorithm, the matching documents are between 0.5-2 % from the total number of documents.

For instance, filtering based on topic 101 (first topic from the English set), the final results are:

- 178 hits in English, 2004
- 102 hits in English, 2005
- 92 hits in English, 2006
- 295 hits in French, 2004
- 167 hits in French, 2005
- 188 hits in French, 2006

So, from a total of 372 hits for English and 650 hits for French are 1022 hits in total out of 200.000 documents, so the percent represent by number of hits is 0,511%.

3 Results

We submitted 4 runs, one run Eng/Eng (with English as source language and target language) and three runs Eng/Eng-Fre (with English as source language and with English and French as target languages). Details from official evaluation are presented below:

Table 1: Number of documents retrieved by runs

Run ID	Source/Target Languages	Number of Documents		
		Retrieved	Relevant	Relevant retrieved
Run 1	Eng/Eng	71.285	1.597	1.331
Run 2	Eng/Fre	66.551	2.421	1.614
	Eng/Eng	71.285	1.597	1.331
	Eng/Eng-Fre	137.836	4.018	2.945
Run 3	Eng/Eng	75.915	1.597	1.507
	Eng/Fre	67.124	2.421	1.905
	Eng/Eng-Fre	143.039	4.018	3.412
Run 4	Eng/Eng	33.793	1.597	1.267
	Eng/Fre	21.591	2.421	1.120
	Eng/Eng-Fre	55.384	4.018	2.387

For Run 1 and Run 2, we used different priorities for the search terms (for example: *Country* tag had the priority 5, *City* tag had 3, *Headline* tag had 7, and *Slugline* tag had 4). For Run 4 we also used priorities for the terms but we looked only in *Headline* and *DataContent* fields. All terms had the standard priority only in Run 3, which returned the best result. For the last three runs we used a translation algorithm to return Eng-Fre results.

The best result was obtained for Run 3, where English was considered as source and as target language. (See Figure 2 and Figure 3).

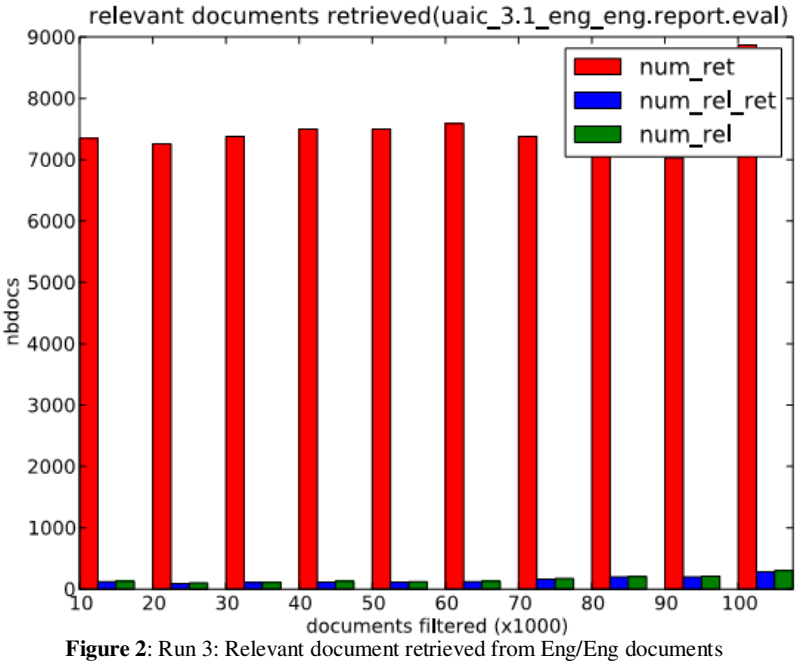

Figure 2: Run 3: Relevant document retrieved from Eng/Eng documents

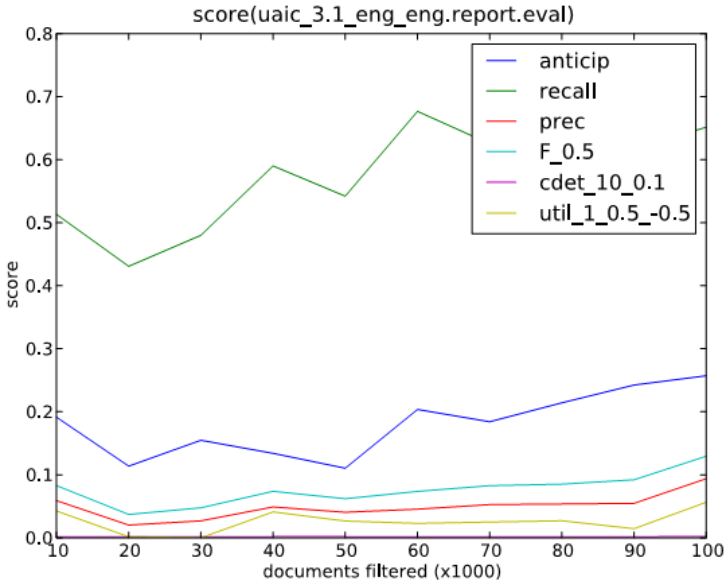

Figure 3: Run 3: Score for Eng/Eng documents

4 Conclusions

This paper presents the UAIC system which took part in the INFILE@CLEF 2009 competition. This year is the second edition of the INFILE campaign and our first participation on this project.

We designed a system formed of several modules: the *parsing module*, which retrieves the relevant content from the documents, the *indexing module* done with Lucene and the *filtering module* which generates a Lucene query and extract from Lucene index the relevant documents.

We submitted 4 runs, and our best result was obtained in Run 3, where English was both the source and the target language.

Acknowledgements

We also like to give a special "thank you" to those who helped from the very beginning of the project: our colleagues from second year group 4 B.

References

1. Besançon R., Chaudiron S., Mostefa D., Hamon O., Timimi I., Choukri K.: Overview of the CLEF 2008 INFILE Pilot Track. *In Working Notes of the Cross Language Evaluation Forum (CLEF 2008)*, Aarhus, September. (2008)
2. Hatcher, E. and Gospodnetic, O.: Lucene in action. *Manning Publications Co*. (2005)