# Decomposing Text Processing for Retrieval: Cheshire tries GRID@CLEF

Ray R. Larson
School of Information
University of California, Berkeley, USA
ray@sims.berkeley.edu

**Abstract**

This short paper is a work in progress describing our participation in the GRID@CLEF task. The GRID@CLEF task is intended to capture in XML form the intermediate results of the text processing phases of the indexing process used by IR systems. Our approach was to create a new instrumented version of the indexing program used with the Cheshire II system. Thanks to an extension by the organizers, we were able to submit runs derived from our system.

The system used for this task is a modified version of the Cheshire II IR system, to which output files for the different intermediate streams have been added. The additions, like the original system were written in C. Developing this system required creating parallel modules for several elements of the Cheshire II indexing programs. The current version handles the simplest processing cases, and currently ignores the many specialized indexing modes in the system (such as geographic name extraction and georeferencing).

## Categories and Subject Descriptors

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

## General Terms

Algorithms, Performance, Measurement

## Keywords

Cheshire II, Logistic Regression

# 1 Introduction

The Berkeley Cheshire group decided to participate in GRID@CLEF for two primary reasons. The first was that the task goal of separating the processing elements of IR systems and looking at the their intermediate output was interesting. The second was more concerned with a detailed reanalysis of our existing processing system and the hope of finding new and better ways to do some of the things that we have developed over the past decade. Since one goal of the GRID@CLEF task is for systems to be able to both export and import intermediate processing streams and eventually to share them, we also hope to be able to use others' streams as inputs for subtasks in which we currently cannot do or cannot do effectively (such as decompounding German words).

The system that we used for GRID@CLEF is a modified version of the Cheshire II IR system, which we have used for all of our participation in various CLEF tracks over the past several years. The modifications made to the system (for this year) primarily concerned the pre-processing and "normalization" of text. In the current implemention of the GRID-enabled system the indexing program is primarily affected. Essentially the indexing program retains all of the functionality that it previously had, but now it will generate output XML files for the different intermediate streams during the text processing and normalization process. These additions, like the original system were written in C. Developing the modified system required creating parallel modules for several elements of the Cheshire II indexing program. Those modules needed to pass along data from a higher level in the call tree down to the low-level code where functions were called to output tokens, stems, etc. to the appropriate files. There are a myriad of alternative parsing approaches, etc. controlled by Cheshire II configuration files, and in this first-cut version for GRID@CLEF only a very few of the most basic ones are supported. Because the system developed over time to support a variety of speciallized index modes and features (such as extracting and georeferencing place names from texts to permit such things as geographic searching though proximity, and extracting dates and times in such a way that they can be searched by time ranges, etc. instead of treating dates as character strings). For the current implementation of we deal only with text extraction and indexing, and do not even attempt to deal with separate indexes for different parts of the documents.

## 2 Information Retrieval Approach

*Note that this section is virtually identical to one that appears in our papers from previous CLEF participation and appears here for reference only[8, 7]*

For retrieval in the GRID@CLEF track we used the same algorithms that we used in other CLEF participation (including for Adhoc-TEL this year), without change. In fact, the basic processing captured by the output files submitted for this track has been fairly standard for our participation across all tracks in CLEF. For retrieval, we used the inverted file and vector file indexes created during the indexing process using the same Logistic Regression-based ranking algorithm that we have used elsewhere.

The basic form and variables of the *Logistic Regression* (LR) algorithm used for all of our submissions was originally developed by Cooper, et al. [5]. As originally formulated, the LR model of probabilistic IR attempts to estimate the probability of relevance for each document based on a set of statistics about a document collection and a set of queries in combination with a set of weighting coefficients for those statistics. The statistics to be used and the values of the coefficients are obtained from regression analysis of a sample of a collection (or similar test collection) for some set of queries where relevance and non-relevance has been determined. More formally, given a particular query and a particular document in a collection $P(R \mid Q, D)$ is calculated and the documents or components are presented to the user ranked in order of decreasing values of that probability. To avoid invalid probability values, the usual calculation of $P(R \mid Q, D)$ uses the "log odds" of relevance given a set of $S$ statistics, $s_i$, derived from the query and database, such that:

$$\log O(R \mid Q, D) = b_0 + \sum_{i=1}^{S} b_i s_i \tag{1}$$

where $b_0$ is the intercept term and the $b_i$ are the coefficients obtained from the regression analysis of the sample collection and relevance judgements. The final ranking is determined by the conversion of the log odds form to probabilities:

$$P(R \mid Q, D) = \frac{e^{\log O(R|Q,D)}}{1 + e^{\log O(R|Q,D)}} \tag{2}$$

## 2.1 TREC2 Logistic Regression Algorithm

For Adhoc-TEL we used a version the Logistic Regression (LR) algorithm that has been used very successfully in Cross-Language IR by Berkeley researchers for a number of years[3]. The formal definition of the TREC2 Logistic Regression algorithm used is:

$$\begin{aligned}
\log O(R|C,Q) &= log\frac{p(R|C,Q)}{1-p(R|C,Q)} = log\frac{p(R|C,Q)}{p(\overline{R}|C,Q)} \\
&= c_0 + c_1 * \frac{1}{\sqrt{|Q_c|}+1}\sum_{i=1}^{|Q_c|}\frac{qtf_i}{ql+35} \\
&+ c_2 * \frac{1}{\sqrt{|Q_c|}+1}\sum_{i=1}^{|Q_c|}\log\frac{tf_i}{cl+80} \\
&- c_3 * \frac{1}{\sqrt{|Q_c|}+1}\sum_{i=1}^{|Q_c|}\log\frac{ctf_i}{N_t} \\
&+ c_4 * |Q_c|
\end{aligned} \tag{3}$$

where $C$ denotes a document component (i.e., an indexed part of a document which may be the entire document) and $Q$ a query, $R$ is a relevance variable,

$p(R|C,Q)$ is the probability that document component $C$ is relevant to query $Q$,

$p(\overline{R}|C,Q)$ the probability that document component $C$ is *not relevant* to query $Q$, which is 1.0 - $p(R|C,Q)$

$|Q_c|$ is the number of matching terms between a document component and a query,

$qtf_i$ is the within-query frequency of the $i$th matching term,

$tf_i$ is the within-document frequency of the $i$th matching term,

$ctf_i$ is the occurrence frequency in a collection of the $i$th matching term,

$ql$ is query length (i.e., number of terms in a query like $|Q|$ for non-feedback situations),

$cl$ is component length (i.e., number of terms in a component), and

$N_t$ is collection length (i.e., number of terms in a test collection).

$c_k$ are the $k$ coefficients obtained though the regression analysis.

If stopwords are removed from indexing, then $ql$, $cl$, and $N_t$ are the query length, document length, and collection length, respectively. If the query terms are re-weighted (in feedback, for example), then $qtf_i$ is no longer the original term frequency, but the new weight, and $ql$ is the sum of the new weight values for the query terms. Note that, unlike the document and collection lengths, query length is the "optimized" relative frequency without first taking the log over the matching terms.

The coefficients were determined by fitting the logistic regression model specified in $\log O(R|C,Q)$ to TREC training data using a statistical software package. The coefficients, $c_k$, used for our official runs are the same as those described by Chen[1]. These were: $c_0 = -3.51$, $c_1 = 37.4$, $c_2 = 0.330$, $c_3 = 0.1937$ and $c_4 = 0.0929$. Further details on the TREC2 version of the Logistic Regression algorithm may be found in Cooper et al. [4].

### 2.2 Blind Relevance Feedback

In addition to the direct retrieval of documents using the TREC2 logistic regression algorithm described above, we have implemented a form of "blind relevance feedback" as a supplement to the basic algorithm. The algorithm used for blind feedback was originally developed and described by Chen [2]. Blind relevance feedback has become established in the information retrieval community due to its consistent improvement of initial search results as seen in TREC, CLEF and other retrieval evaluations [6]. The blind feedback algorithm is based on the probabilistic term relevance weighting formula developed by Robertson and Sparck Jones [9].

Blind relevance feedback is typically performed in two stages. First, an initial search using the original topic statement is performed, after which a number of terms are selected from some number of the top-ranked documents (which are presumed to be relevant). The selected terms are then weighted and then merged with the initial query to formulate a new query. Finally the reweighted and expanded query is submitted against the same collection to produce a final ranked list of documents. Obviously there are important choices to be made regarding the number of top-ranked documents to consider, and the number of terms to extract from those documents. For ImageCLEF this year, having no prior data to guide us, we chose to use the top 10 terms from 10 top-ranked documents. The terms were chosen by extracting the document vectors for each of the 10 and computing the Robertson and Sparck Jones term relevance weight for each document. This weight is based on a contingency table where the counts of 4 different conditions for combinations of (assumed) relevance and whether or not the term is, or is not in a document. Table 1 shows this contingency table.

Table 1: Contingency table for term relevance weighting

| | Relevant | Not Relevant | |
|---|---|---|---|
| In doc | $R_t$ | $N_t - R_t$ | $N_t$ |
| Not in doc | $R - R_t$ | $N - N_t - R + R_t$ | $N - N_t$ |
| | $R$ | $N - R$ | $N$ |

The relevance weight is calculated using the assumption that the first 10 documents are relevant and all others are not. For each term in these documents the following weight is calculated:

$$w_t = log \frac{\frac{R_t}{R - R_t}}{\frac{N_t - R_t}{N - N_t - R + R_t}} \tag{4}$$

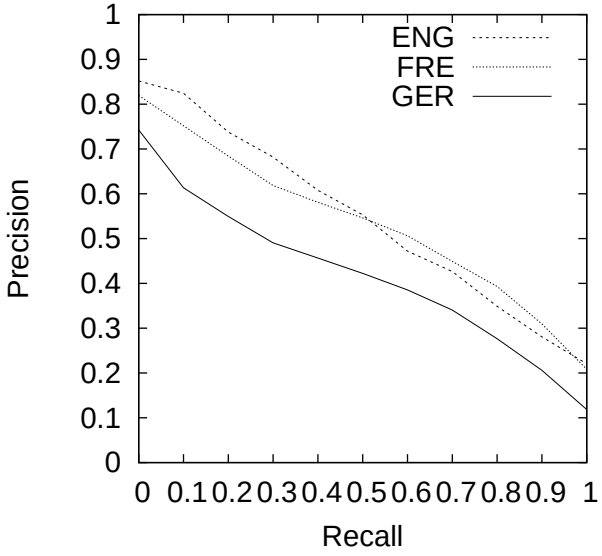
The 10 terms (including those that appeared in the original query) with the highest $w_t$ are selected and added to the original query terms. For the terms not in the original query, the new "term frequency" ($qtf_i$ in main LR equation above) is set to 0.5. Terms that were in the original query, but are not in the top 10 terms are left with their original $qtf_i$. For terms in the top 10 and in the original query the new $qtf_i$ is set to 1.5 times the original $qtf_i$ for the query. The new query is then processed using the same LR algorithm as shown in Equation 4 and the ranked results returned as the response for that topic.

## 3 Text Processing Result Submissions

For GRID@CLEF in addition to the conventional retrieval runs (described in the next section), we submitted four intermediate streams from the indexing process. These were:

**Basic tokens** – in Cheshire II parsing into tokens takes place once an XML sub-tree of a document required for a particular index specified in a configuration file is located. To keep things as simple as possible in this version, the XML sub-tree is the entire document (e.g., the <doc> tag and all of its descendents). Tokenization first eliminates all XML tags in the subtree (replacing them with blanks) and then uses the "strtok" C string library function to include

Figure 1: Berkeley Monolingual Runs, English, French, and German




any sequence of alphanumeric characters divided at white space or punctuation (with the exception of hyphens and periods, which are retained at this point). Hyphens are treated specially and double extracted, once as the hyphenated word and then as separate words with the hyphen(s) removed. (At least that is what it SHOULD be doing – in checking results for this stage I found that only the first word of a hyphenated word was getting extracted. This is now being corrected). Sequences of letters and periods are assumed to be initialisms (like U.S.A.) and are left in the basic token stream.

**Lowercase normalization** – The default normalization (which can be turned of by the configuration files) is to change all characters to lowercase. This step also removes any trailing period from tokens (so U.S.A. becomes u.s.a).

**Stopword removal** – Each index can have an index-specific stoplist and any words matching those in the stoplist are thrown out and don't go on to any later stages.

**Stemming** – For each collection the configuration file specified use of particular stemmers including the Snowball stemmer for various languages and an extended version of the Porter stemmer. The Snowball stemming system has been integrated into the Cheshire II system and any of its stemmers can be invoked via different configuration file options.

Finally the remaining stemmed tokens are accumulated along with their document frequency information and stored in a temporary file. In subsequent stages the information for all of the documents is sorted, merged and an inverted file created from the tokens and their document frequency information.

In retrieval, the same stages are performed on the tokens derived from the topics or queries before matching takes place.

The XML files produced for each of these streams ranged in size from 18.5Gb for raw token files to 4.5Gb after stemming, depending on the test collection and the position in processing.

# 4 Retrieval Results

Although our retrieval runs were submitted quite late, the organizers kindly allowed them to go through the same evaluation as the officially submitted runs. We submitted only one monolingual

run for each of English, French, and German.

The indexes and vector files created during the later stages of the indexing process (and not yet captured by the GRID@CLEF output streams) were used to provide the matching used in the logistic regression algorithm described above. Overall, the retrieval results look fairly good (although there was only one other participant to compare with) with comparable results in all languages (except German, where I suspect the other group is using decompounding).

Figure 1 shows the precision-recall graph for all of our submitted runs. The MAP of our German run was the lowest at 0.4003, with a MAPs of 0.5313 and 0.5188 for English and French, respectively. Interestingly, the identical algorithm and processing (without capturing the intermediate outputs) was used in our Adhoc-TEL participation this year, with much worse performance in terms of average precision when compared to even the same group also participating in this task, which shows that the same algorithms and processing systems can have radically different performance on different collections and query sets.

# 5 Conclusions

One of the goals in our participation in GRID@CLEF was to identify problems and issues with our text processing and normalization stages. In that we have been quite successful, having identified one definite bug and a number of areas for re-design and enhancement. The next phase would be to enable the system to take any of the intermediate streams produced by different participants as input. This is a much more difficult problem, since much further work and analysis is needed. Since, for example the Cheshire system can create separate indexes based on different parts of an XML or SGML record, the streams would also need to carry this kind of information along with them. In addition, some of our indexing methods perform the text processing in different sequences (for example, geographic name extraction uses capitalization as one way of identifying proper nouns that might be place names, and the output of the georeferencing process is a set of geographic coordinates instead of a text name).

Overall this has been a very interesting and useful track and provided several improvements to our system that will carry over to other tasks as well.

# References


[1] Aitao Chen. Multilingual information retrieval using english and chinese queries. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF-2001, Darmstadt, Germany, September 2001*, pages 44–58. Springer Computer Scinece Series LNCS 2406, 2002.

[2] Aitao Chen. *Cross-Language Retrieval Experiments at CLEF 2002*, pages 28–48. Springer (LNCS #2785), 2003.

[3] Aitao Chen and Fredric C. Gey. Multilingual information retrieval using machine translation, relevance feedback and decompounding. *Information Retrieval*, 7:149–182, 2004.

[4] W. S. Cooper, A. Chen, and F. C. Gey. Full Text Retrieval based on Probabilistic Equations with Coefficients fitted by Logistic Regression. In *Text REtrieval Conference (TREC-2)*, pages 57–66, 1994.

[5] William S. Cooper, Fredric C. Gey, and Daniel P. Dabney. Probabilistic retrieval based on staged logistic regression. In *15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, June 21-24*, pages 198–210, New York, 1992. ACM.

[6] Ray R. Larson. Probabilistic retrieval, component fusion and blind feedback for XML retrieval. In *INEX 2005*, pages 225–239. Springer (Lecture Notes in Computer Science, LNCS 3977), 2006.

[7] Ray R. Larson. Cheshire at geoclef 2007: Retesting text retrieval baselines. In *8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, LNCS 5152, pages 811–814, Budapest, Hungary, September 2008.

[8] Ray R. Larson. Experiments in classification clustering and thesaurus expansion for domain specific cross-language retrieval. In *8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, LNCS 5152, pages 188–195, Budapest, Hungary, September 2008.

[9] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, pages 129–146, May–June 1976.