Universities of Avignon & Lyon III at TREC 2008: Enterprise Track

Eric SanJuan[*]
LIA & IUT STID, University of Avignon, FRANCE

Nicolas Flavier[†]
LIA & CERI, University of Avignon, FRANCE

Fidelia Ibekwe-SanJuan[‡]
ELICO, University of Lyon 3, FRANCE

Patrice Bellot[§]
LIA & CERI, University of Avignon, FRANCE

1 Introduction

The Enterprise track of TREC 2008 comprised of the same two tasks as in the previous years: an ad-hoc document search and an expert search.

- The document search consisted in retrieving documents that best matched real-life queries submitted by users to the CSIRO corporation. Systems were allowed to retrieve and rank up to a 1000 documents.
- The expert search consisted in locating the CSIRO staff who is best able to respond to the query formulated by the users.

This year was our first participation in TREC-ENT.

We explored three major approaches to information retrieval using various existing methods and systems. These approaches ranged from domain knowledge mapping [2] to QA [1].

[*] eric.sanjuan@univ-avignon.fr
[†] nicolas.flavier@univ-avignon.fr
[‡] ibekwe@univ-lyon3.fr
[§] patrice.bellot@univ-avignon.fr

2 Document search

Three document runs were submitted in this task. Each run tested a different search methodology, ranging from SOMs using a general ontology, to question-answering and passage retrieval, and then to manual query expansion based on relevance feedback.

2.1 General ontologies for knowledge organization and domain mapping based on Self Organized Maps

Two runs were carried out using this strategy.

- In **LiaIIcAuto** run, a small set of documents was extracted and concatenated using **Lemur** and the query title field.
- In **LiaIcAuto** run, all query fields were concatenated.
- In both cases, resulting texts were projected onto the knowledge map previously built on the whole data. Documents were then ranked by similarity. The runs were completely automatic. There was no human intervention on the ontology.

2.2 Question-Answering based on SIAC4QA segmenter

This run was also completely automatic.

Question Answering systems aim at retrieving precise answers to questions expressed in natural language. Questions are mainly factual questions and answers are pieces of text extracted from a collection (such as newspaper article compilation). They have been particularly studied since 1999 and the first large scale QA evaluation campaign held as a track of the Text REtrieval Conference.

Typical QA system architecture involves at least these main steps (most often pipelined):

- Question Analysis, to extract semantic type(s) of the expected answer;
- Document Retrieval to restrict the amount of processed data by further components;
- Passage Retrieval to choose the best answering passages from documents;
- and final Answer Extraction Strategies to determine the best answer candidate(s) drawn from the previously selected passages.

We employed the Passage Retrieval component in TREC Enterprise as an Indri post-processing. Applied to TREC Enterprise data, the inputs are the title fields of the topics and the sets of documents, and the outputs are some ranked lists of retrieved passages.

Since our first TREC QA participation [1], our passage retrieval approach changed from a cosine based similarity to a density measure. For QA, our passage retrieval component sees a question as a set of several kinds of items : words, lemmas, POS tags, Named Entity tags, and expected answer types. For experiments, items were the lemmas of the topic only (the empty words were filtered according to their POS tags) and the maximum size of a retrieved passage has been limited to three sentences.

First, a density score s is computed for each occurrence o_w of each topic lemma w in a given document d. This score measures how much the words of the topic are far away from the other ones. It allows to point at the centers of the document areas where the words of the topic are most present. It takes into account the number of different lemmas $|w|$ in the topic, the number of topic lemmas $|w, d|$ occurring in the currently processed document d and a distance $\mu(o_w)$ that equals the average number of words from o_w to the other topic lemmas in d (in case of multiple occurrences of a lemma, only the nearest occurrence to o_w is considered).

Let $s(o_w, d)$ be the density score of o_w in document d:

$$s(o_w, d) = \frac{\log\left[\mu(o_w) + (|w| - |w, d|)\,.p\right]}{|w|}$$

where p is an empirically fixed penalty aimed to prefer or to not prefer few common words with the topic that are close to each other or many words that are distant to each other.

Secondly, a score is computed for each sentence S in a document d. The score of a sentence is the maximum density score of the topic lemmas it contains:

$$s(S, d) = \max_{o_w \in S} s(o_w, d)$$

At the end of the process, the score of a document is the linear combination of the original INDRI score with the passage retrieval score. This resulted in the in the **LIAIndriSiac** run.

2.3 Multiword term incremental query expansion using relevance feedback

From the observation that the topics in TREC-ent were real life complex queries that would normally involve humans somewhere in the loop in order to “construct” the answer. Indeed, a manual inspection showed that often, the answer was not readily available on the retrieved web pages. It needed to be “constructed” from reading several potentially relevant web pages. Topics of the type “How can I do Y about X?” would typically have pages containing some information about X but not necessarily the real answer (“how to do Y”).

These topics particularly relevance feedback techniques in order to expand the queries with more adequate terms. The query expansion strategy consisted in submitting an initial query to Indri using terms from the title field. Additional multiword terms were manually gathered from an exploration of the top 20 documents ranked by Indri. These terms were then used to expand the initial set of query terms and resubmitted to Indri. The final set of query terms was submitted to the **Indri engine** using:

- proximity operators (#3)

- belief operators (#combine).

This run is named **LiaIndriMan**.

2.4 Preliminary results for document search

For this corpus our baseline run consisted in submitting the content of the title field to Indri. This baseline attained a high average score: infAP=0.3167, infNDCG=0.5008 on queries. We observed a similar performance TREC-ENT 2007 data. Only the **LIAIndriSiac** run attained a higher average score: infAP=0.3191 infNDCG=0.5078.

The run **LiaIndriMan** using manually expanded multiword terms obtained a quite lower score: infAP=0.2379, infNDCG=0.3951. This score can be improved by relaxing the NP structure of the multiword terms and allowing the insertion of more words into MTW (We added 2 to all #n indri operators). It is also improved using automatic query expansion. The manual run finally obtains the following average scores: infAP=0.2734, infNDCG=0.4461, that still remain under the baseline.

The average score of other two runs are even lower (under 0.1 for infAP and 0.2 for infNDCG). This could be explained by the gap between the knowledge base we used (specialized scientific domains and economic vocabulary) and the common vocabulary in CSIRO web pages.

However, when we look at the performance of our runs, query by query, we find out that each system works better on some type of query. Figure 1 shows the results query by query. The left bar represents the median score of all participants. It clearly appears that LIAIndriSiac is often over the median score but when the median score is low, then the run based on manually extracted terms performs better.

3 Expert search

We carried out a baseline search using Indri and a manual search.

3.1 Automatic baseline run

This consisted in generating multi-document summaries for each e-mail address occurring in the corpus. These summaries were indexed using Lemur and addresses were ranked based on indri #combine operator applied to titles without any preprocessing. This run is labeled **LiaExp08**.

3.2 Manual Run

RunID: **LiaIcExp08**. This run was carried out following these steps:

1. creating an expert sub-collection using the query terms “dr” and “professor” in html title fields.
2. an automatic search was then done by similarity of concepts with query and narrative fields just copied into the search mask. Concept similarity relies on a general ontology and a domain map built on the sub-collection.

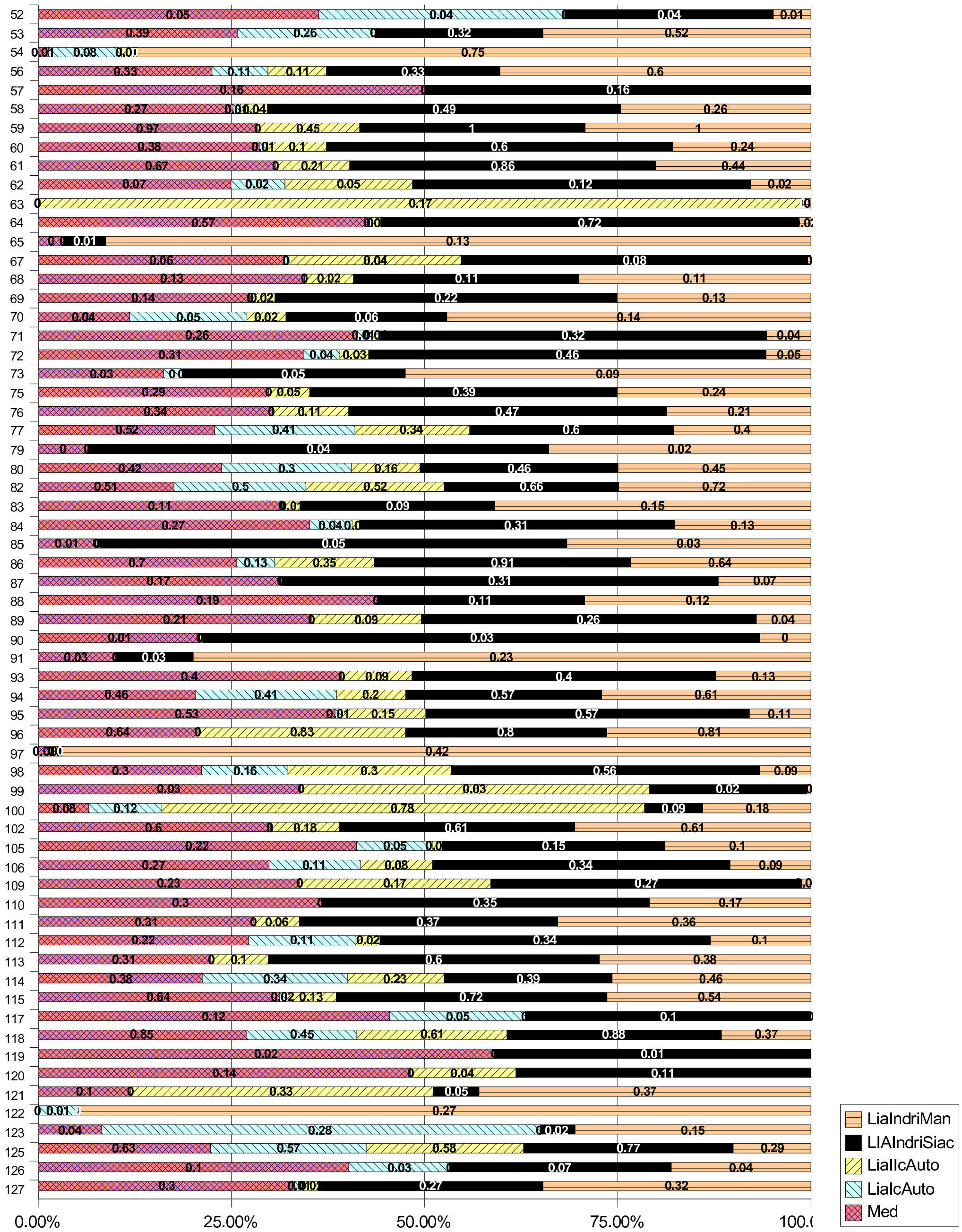

Figure 1: Inferred Average Precision for Lia runs and TREC 2008 median score on document search

- When relevance was above a user defined threshold, documents were opened for selection and/or query refinement.
- Otherwise the query was run on the entire corpus and same process using the corresponding larger domain map.

3. If relevance was again too low, the query was reduced to traditional keyword list by deletion of meaningless words. Then a search by synonyms was applied.

3.3 Results

It appeared that the user did not find more than four experts per query with an average of 2.44. This is in contrast with the resulting qrels established by participants where there is an average of 10.36 experts per query. Therefore the map score of LiaIcExp08 is only 0.2513. However, ircl_prn.0.00 is 0.8576 and ircl_prn.0.10 is 0.7806 in average on all queries.

Still, even on these qrels, the manual run significantly outperforms our baseline that has a map score of 0.1841 with ircl_prn.0.00=0.5906 and ircl_prn.0.10=0.5393.

References

[1] P. Bellot, E. Crestan, M. El-Bèze, L. Gillard, and C. de Loupy. Coupling named entity recognition, vector-space model and knowledge bases for trec-11 question-answering track. In *The Eleventh Text REtrieval Conference (TREC 2002), NIST Special Publication 500-251*, 2003.

[2] F. Ibekwe-SanJuan and E. SanJuan. From term variants to research topics. *Journal of Knowledge Organization (ISKO), special issue on Human Language Technology*, 29(3/4), 2003.