# WHU at TREC Total Recall Track 2015

Chuan Wu
School of Information Management
Wuhan University
Wuhan, Hubei, China
wu.chuan@whu.edu.cn

Wei Lu
School of Information Management
Wuhan University
Wuhan, Hubei, China
weilu@whu.edu.cn

Ruixue Wang
School of Information Management
Wuhan University
Wuhan, Hubei, China
ruixue_wang@whu.edu.cn

## ABSTRACT

This paper describes the WHU IRLAB participation to the Total Recall Track in TREC 2015. We implement an end-to-end system to deal with the total recall task. We propose an iterative query expansion method, which construct queries using iteratively selected terms. We choose to participate the "Play-at-home" evaluation. Results are presented and discussed.

## Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Abstracting methods

## Keywords

keywords

## 1. INTRODUCTION

The Text Retrieval Conference (TREC) this year introduces a new track called Total Recall Track. Given a set of topics as queries, and a collection of documents, participants are required to find as many relevant documents as possible with few effort. High recall is the primary concern. However, since it does not make sense if the cost of information seeking itself is too much, another goal is to limit effort paid into the information seeking behavior.

In Total Recall Track, the relevance of documents with respect to topics are stored in remote server. All documents identified as relevant are submitted to server for relevance judgments, and the number of times issuing a query against the index is regarded as an effort.

Two kinds of evaluation are provided, i.e. "Play-at-home" evaluation and "Sandbox" evaluation. For "Play-at-home", participants ran their system on their own with the choice of "automatic" and "manual", indicating whether manual intervention is included. For "Sandbox" evaluation, a virtual machine with a fully automated solution is submitted.

Based on our limitation, we focus on "Play-at-home" evaluation. We propose an iterative query expansion approach to find relevant documents. In order to achieve this goal, we need to first retrieve relevant documents, and then find subtopics about the given information need. Basically, this is an iterative process as we can always find something new in retrieved relevant documents until all relevant documents are found.

The rest of this paper is organized as follows. Section 2 gives a brief introduction about the Total Recall task. Section 3 presents the whole framework of our Total Recall System. Section 3 describes how we use query expansion techniques to resolve Total Recall. Section 4 describes details about our experimentation. We conclude in Section 5.

## 2. METHOD

In this section we present the framework used to resolve total recall task. We describe all necessary preprocessing steps, followed by our iterative query expansion method.

### 2.1 Preprocessing

As required by Total Recall Track organizers, automatic experiments must use software that, without human intervention, downloads the dataset and conducts the task end to end. Our preprocessing process includes three steps: Downloading Corpus, Corpus Preprocessing, Index Construction.

First of all, we get a runid from total recall server, and then information of the corpus corresponding to the runid can be obtained, including the url of the corpus. Then the corpus is downloaded and uncompressed. Second, we perform some preprocessing on the corpus if necessary. After analyzing some corpus file, we found that some xml file cannot be directly handled by XML parsing tools. Therefore, we adjust the xml files to make the files well-formed. Finally, we construct the index of the corpus.

### 2.2 Query Expansion based approach

Given a topic and a collection of documents, our goal is to iteratively find distinctive terms from retrieved documents and expand the original query using these terms. Our approach consists of two iterative steps, i.e. document retrieval, and distinctive term selection. By first retrieving documents using the original query or expanded query and obtaining relevance judgments from server, we find relevant documents. Then we identify distinctive terms from relevant documents to capture various aspects of the query. The whole process is illustrated as follows:

1) Set query $Q_{00}$ , the first query in the first iteration as the original query $Q$.

2) In the *i-th* iteration, we have *m* queries to perform query expansion.

3) For the *j-th* query in the *i-th* iteration, search the given corpus using $Q_{ij}$ to get top $N$ retrieval result set.

4) Obtain relevance judgments from Total Recall Track Server. All relevant documents are grouped into the Relevant Result Set (RRS), which is denoted as $RRSij$.

5) If $RRS_i$ is empty, then this branch ends, otherwise extract all distinctive terms from $RRS_{ij}$ and denote these terms as Distinctive Term Set (DTS), i.e. $DTS_{ij}$;

6) If $DTS_{ij}$ is empty, then this branch ends, otherwise construct $|DTS_{ij}|$ queries using $DTS_{ij}$ by query formula $Q + w_i$. Go to step 2.

### 2.2.1 Document Retrieval

In document retrieval step, we first search the given corpus using the original query or an expanded query and obtain a ranked list of documents, denoted as Result Set (RS). Instead of submitting all retrieved documents to the server for relevance judgments, we submit top N documents. The reason is that the top results are assumed to be more likely to be relevant. If the number of retrieved document is less than N, then we submit all retrieved documents.

After obtaining the relevance judgments from the server, we have two sets of documents, i.e. *relevant document set (RDS)* and irrelevant document set. We pass *RDS* to the next step to extract distinctive terms for the next round of query expansion if *RDS* is not empty. If *RDS* is empty, which means that no relevant documents is found for the given query, move the next query.

### 2.2.2 Distinctive Term Selection

Since how a document is relevant with a query is encoded in distinctive terms in the document, we propose to identify distinctive terms from relevant documents and perform query expansion using these terms. We implement the Inverse Local Context Analysis method with some modifications.

Given a query Q and a relevant document set S, identify distinctive terms from S. The steps are as follows:

1) Iterate over all terms in all documents, for each term $t_i$,

2) Rank terms within $RRS_i$ using $f(c, QS)$ (see Formula 1) in ascent order.

$$f = (c, Q) = \prod_{w_j \in Q} \lambda + co\_degree(c, w_j))^{idf(w_j)}$$

$$co\_degree(c, w_j) = |D \in RRS_i | c \in D, w_j \in D|$$

3) Select the top k terms in the ranked term list.

Ranking terms in ascent order means that the lower the relatedness of a term with query terms, the higher the term will be ranked. Then we select top K terms as candidate expansion terms. For each expansion term, we go to step 1 for the next round of query expansion.

# 3. EXPERIMENT

## 3.1 Experimental setup

The document collection, information need and relevance assessor are all supplied to participants via an on-line server. For the "Play-at-home" evaluation, three tests are provided, i.e. athome1, athome2, and athome3. For each test, a specific corpus and corresponding topics are provided. We apply our method on athome1. The details about the datasets in each test are shown in Table 1.

## 3.2 Evaluation metrics

As stated by TREC Total Recall Track organizers, both "completeness" and "effort" are reflected in evaluation metrics. Completeness means how nearly all of the relevant documents are found, while effort is a function of the number of documents submitted to the assessment server. In order to evaluate completeness and effort in total recall, Two kinds of evaluation metrics are given, i.e. Rank measures and Set measures. Rank measures reflect completeness for various effort values, while Set measures reflect completeness at a fixed level of effort. Details about the evaluation metrics can be found in Total Recall website.

| | #Rel. | Max. recall | Effort | Precision |
|---|---|---|---|---|
| athome101 | 5836 | 0.0019 | 13 | 0.8462 |
| athome102 | 1624 | 0.0062 | 11 | 0.9091 |
| athome103 | 5725 | 0.0690 | 395 | 1.0000 |
| athome105 | 3635 | 0.0561 | 204 | 1.0000 |
| athome106 | 17135 | 0.0770 | 1319 | 1.0000 |
| athome107 | 2375 | 0.0497 | 118 | 1.0000 |
| athome108 | 2375 | 0.1158 | 276 | 0.9964 |

**Table 1: Precision, recall of our Method**

## 3.3 Results

On the athome1 dataset, the recall is quite limited compared to the provided baseline. Given that our effort is quite limited, our iterative process ends earlier than expected. The reason could be that we didn't find appropriate combination distinctive terms for query expansion. Simply by adding one distinctive term to perform query expansion is not enough to find all relevant documents. The high precision is reasonable since no much documents are retrieved. It is not very difficult to identify relevant documents in the first rounds.

# 4. CONCLUSION

We participated in the newly introduced TREC 2015 Total Recall Track. One run was submitted for "Play-at-home" evaluation. We proposed an iterative query expansion approach to improve total recall. The results show that the performance of our simple query expansion approach is not as good as the provided baseline. It might be important to find appropriate combination of terms for query expansion.

# 5. ACKNOWLEDGMENTS

This work is supported by the National Social Science Fund of China (Grant No. 12&ZD1221) and the National Natural Science Foundation of China (Grant No. 71173164).