York University at TRECVID 2010

Jun Miao[1], Xiangji Huang[1], Qinmin Hu[1]
[1] Information Retrieval and Knowledge Managment Lab, York University, Toronto, Canada
jun@cse.yorku.ca, jhuang@yorku.ca, vhu@cse.yorku.ca

Abstract

In this paper, we describe our work done by members at York University in Canada for the KIS (Known-item search) task of TRECVID 2010. This is the first time that we participate in the TRECVID. With rich experience in text retrieval, we mainly focus on the meta information of videos, and try to figure out the importance of these description corpus. In order to obtain this goal, we do not use any video or audio technologies. Only text retrieval methods are utilized to find the know items. Traditional weighting models in text retrieval like BM25 and Lemur TF-IDF are used. Meanwhile, we also use query expansion methods to improve the performance. However, the results are not promising. We make a further discussion about the reason at the end of this paper.

Keywords

Video Track, Know Item Search, Mean Average Precision, BM25, Lemur TF-IDF, KL, Bo1

1 Introduction

It is the first time that we participat in the TREC Video Retrieval Evaluation (TRECVID). The main goal of our participation is to find a efficient method for the Known-item Search (KIS) task. The KIS task supposes the situation in which a person knows the content of a video but he doesn't know where to obtain it. He or she then inputs a text-based description as a query to find the video [5] [6]. The test data set for this task contains about 8000 Internet archive videos with associated metadata. Since we have plenty of experiences in text retrieval, it is intuitively for us to evaluate the importance of all the metadata which are textual.

A typical meta file is organized as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata>
  <collection>bliptv</collection>
  <mediatype>movies</mediatype>
  <resource>movies</resource>
```

```
  <title>Nightmare at WebU</title>
  <description>As part of my role as headmaster and mug washer at my newspaper
  chain&amp;apos;s WebU - a digital and web tech training facility - I&amp;apos;ve
  decided to produce a weekly short video welcome to the students, one shot entirely
  in the morning of their first day of their week-long sojourn at the school. This
   horror movie spoof was Week Four&amp;apos;s welcome</description>
  <upload_application appid="blip.tv" version="1.0"/>
  <uploader>bill.dunphy@gmail.com</uploader>
  <licenseurl>http://creativecommons.org/licenses/by-nc-nd/2.0/</licenseurl>
  <runtime>00:03:21</runtime>
  <publicdate>2007-11-01 02:57:29</publicdate>
  <identifier>Billdunphy-NightmareAtWebU351</identifier>
</metadata>
```


As we can see, there are some features about a particular video file in the data set. Are they effective for the known-item task? To answer the question, we submit 4 automatic runs: YorkKISrun1-4. Each of them is based on the combination of a traditional weighting model and a query expansion model of text retrieval. Although the results are not good, we can get some meaningful conclusions from them.

The rest of the paper is organized as follows. In Section 2 we review the two weighting models we used for the task. Section 3 describe the two query expansion models. Section 4 shows the details of our experiments and results. Next, we made a discussion about the results in Section 5. Finally, we draw some conclusions and present our future work in the last section.

2 Weighting Models

In our experiments, we apply probabilistic models as the weight functions. Terms are assigned weights based on their frequencies in documents and topics. Documents are ranked according to their probabilities of relevance to the topics. The score of each document is the sum of term weights. We used two well-known weighting models, BM25 [3] and Lemur TF-IDF [7] in this year trecvid.

2.1 BM25

In BM25, search term is assigned weight based on its within-document term frequency and query term frequency [3]. The corresponding weighting function is as follows:

$$w = \frac{(k_1+1)*tf}{K+tf} * \log\frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} * \frac{(k_3+1)*qtf}{k_3+qtf} \\ \oplus \quad k_2 * nq * \frac{(avdl-dl)}{(avdl+dl)} \tag{1}$$

where w is the weight of a query term, N is the number of indexed documents in the data set, n is the number of documents containing a specific term, R is the number of documents known to be relevant to a specific topic, r is the number of relevant documents containing the term, tf is within-document term frequency, qtf is within-query term frequency, dl is the length of the document, $avdl$ is the average document length, nq is the number of query terms, the k_is are tuning constants, K equals to $k_1 * ((1-b) + b * dl/avdl)$, and $\oplus$ indicates that its following component is added only once per document, rather than for each term. In our experiments, the values of k_1, k_2, k_3 and b are empirically set to be 1.2, 0, 8 and 0.75 respectively.

2.2 Lemur TF-IDF

Lemur TF-IDF [7] is a transformation of the traditional TF-IDF. It uses the Okapi TF formula to replace the common term frequency:

$$tf_d = \frac{k_1 * tf}{k_1 * (1 - b + b * \frac{dl}{avdl}) + tf} \tag{2}$$

where tf_d is the new term frequency based on Okapi TF formula in a document, tf is within-document term frequency, dl is the length of the document, $avdl$ is the average document length, the k_1 and the b are tuning constants.

For the term frequency tf_q, we directly uses the occurrences of a term in the query. The IDF (Inverse Document Frequency) function is as follows:

$$idf = log(\frac{n}{N} + 1) \tag{3}$$

where N is the number of indexed documents in the data set, n is the number of documents containing a specific term.

Thus, the weight of a term is calculated as:

$$w = tf_d * idf * tf_q * idf = tf_d * tf_q * idf^2 \tag{4}$$

3 Query Expansion Models

In this section, we first present the Rocchio's query expansion framework. Then, we describe two proposed term weighting models for query expansion under Rocchio's framework.

3.1 Rocchio Query Expansion

The Rocchio relevance feedback [4] is a classic algorithm for relevance feedback. It models a way of incorporating relevance feedback information into the vector space model. In particular, it take a set of documents for feedback. The weights of candidate terms in this set of documents are calculated according to the following formula:

$$Q_1 = \alpha * Q_0 + \beta * \sum_{rel} \frac{D_i}{|D_i|} - \gamma * \sum_{nonrel} \frac{D_i}{|D_i|} \tag{5}$$

where Q_0 and Q_1 represent the initial and first iteration query vectors, D_i represents document weight vectors, $|D_i|$ is the corresponding Euclidian vector length, and α, β, γ are tuning constants.

Many other relevance feedback techniques and algorithms have been developed, most of which are derived under Rocchios framework. G.Amati proposed a relevance feedback algorithm in his Divergence from Randomness (DFR) framework [1], which similarly follows Rocchios algorithm. However, in Amati's method, term weights are assigned by DFR term weighting models, such as the Kullback-Leibler divergence (KLD) [2] and Bo1 [1].

In our experiments, we explore two weighting schemes under Rocchio's framework, and the parameters α, β, γ are empirically set to be 1, 0.4 and 0 respectively. In addition, the number of expansion terms, *exp_term*, is empirically set to be 10 in our experiments. In the following subsection, we describe the algorithms in detail.

3.2 KL Weighting Scheme

Kullback-Leibler divergence is a model described in [2]. The basic idea of this term weighting model for query expansion is to measure the divergence of a term's distribution in a pseudo relevance set from its distribution in the whole data set. The higher this divergence is, the more likely the term is related to the query topic. The weight of a term t in the *exp_doc* top-ranked documents is given by:

$$w = \begin{cases} 0 & tf_{rel} < tf_{coll} \\ tf_{rel} * log_2 \frac{tf_{rel}}{tf_{coll}} & Otherwise \end{cases} \tag{6}$$

where tf_{rel} is the frequency of the term in the *exp_doc* top-ranked documents, tf_{coll} is the frequency of the term in the whole data set. *exp_doc* is set to be 3 in our experiments.

3.3 Bose-Einstein distribution Weighting Scheme

Bo1 is another weighting model in the DFR framework. It is based on the Bose-Einstein statistics. Using this model, the weight of a term t in the *exp_doc* top-ranked documents is given by:

$$w = tf_x * log_2 \frac{1 + P_n}{P_n} + log_2(1 + P_n) \tag{7}$$

where *exp_doc* doc usually ranges from 3 to 10 [1]. Another parameter involved in the query expansion mechanism is *exp_term*, the number of terms extracted from the *exp_doc* top-ranked documents. *exp_term* is usually larger than *exp_doc* [1]. P_n is given by F/N , F is the frequency of the term in the data set, and N is the number of documents in the data set. tf_x is the frequency of the query term in the *exp_doc* top-ranked documents.

4 Experiment

4.1 Meta Files

In the test data set for the KIS task, there are 8383 videos. Each video has a meta file which provides descriptions about the it. 8364 of them are standard in XML structure, while others are HTML files or broken files. There are 83 kinds of information in these XML files. The most frequent ones in the XML files are: *collection, identifier, mediatype, metadata, publicdate, licenseurl, uploader, description, subject*. All of them appear more than 7000 times. These contents are used to build index for our experiments.

4.2 Queries

The 300 topics of the KIS task are textual. Each topic contains two parts: a 1-5 keywords visual cue and a query. The query is a description about the item we retrieve. Terms in the cue may not occur in the query part. Sometimes, a long query is not so good as a keyword-based description because there are some noisy terms. In order to figure out which form of the topics describe the items better, we generate two kinds of queries. The first one which is called keyword-based query only contains terms in the visual cues, and the second one, full-text query, contains both of them.

4.3 Experiment Results

In our experiments, all the 4 runs were automatic. Meanwhile, we did not use the training data to adjust any parameters. As we described previously, two weighting models and two query expansion schemes were utilized. We combined BM25 with Bo1 query expansion scheme, and Lemur TF-IDF with KL scheme. The two combinations are used on both keyword-based and full-text topics. Because topic 8 has a duplicate answer and topic 11 is not made correctly, there are totally 298 valid topics.

	York University Runs	Amount of Items Found
1	BM25 + Bo1 + Keyword-based queries	2
2	BM25 + Bo1 + Full-text queries	3
3	Lemur TF-IDF + KL + Keyword-based queries	3
4	Lemur TF-IDF + KL + Full-text queries	3

The results are not acceptable. In the next section, we will make a in-depth analysis in the next section to figure out why the performance is so disappointing.

5 Discussion

Why the results of our experiments are disappointing? We compared the topics with their corresponding videos and meta files and found some clues.

First of all, half of the meta files of target videos do not contain any terms (excluding stop words) which appear in the topics. Since text retrieval is a process of dealing with "bag of words" in most cases, it is difficult to find the target files when lacking common terms in both topics and meta files.

Secondly, even when some meta files and the topics overlap partially, there are some obstacles to find the targets. Sometimes, only few terms occur in both a topic and its according video meta file. That means, it is very possible that the retrieval process is misled by other noisy terms in the topics. Additionally, although there are a relatively large amount of topic terms occur in some meta files, they are always not important keywords for retrieval. For example, for topic 22 — "Find the video of a man and woman getting dressed, a cat on window sill and another cat joining it, a wedding, two kittens and two babies", a file with many "man" and "dress" in it is not likely to be the meta file of the target video because it loses most of information in the topic. However, it can get a high score in the ranking process because it has more keywords than others. For the KIS task, a target video has to contain all the elements mentioned in the topic. Thus, traditional text retrieval models are not appropriate for the task because they mainly focus on the occurrences of individual terms.

Finally, when comparing the descriptions in meta files with the content of videos, we found that some meta files do not have any information about the items in the videos. For example, a description only presents the video was taken through a window, but we need to find a house which is in this video. Since there are not any words about the house in the meta file, it is reasonable to miss the target.

In the results of our 4 runs, all of them obtained the same 2 items for topic 51 and 141. Surprisingly, the meta files of the target videos only contain few topic words. They are found not because they match the topics very well, but because other files are worse. Thus, we can conclude we can hardly find the certain items by traditional text models on the meta files.

6 Conclusion

In this paper, we present our work in the KIS task of TRECVID 2010. We mainly focus on the aid of meta files in searching known items. In order to know the performances of different models, we

implement the combination of BM25, Lemur TF-IDF and two query expansion models. However, the results are not good. We compare the text corpus in the data set with the topics, and found several drawbacks of obtaining target videos via text retrievals. These drawbacks are discussed in the Section 5. The meta files are not effective in searching know items because most of them do not contain the information for the topics. Thus, we conclude that meta files cannot help us much in the KIS task so far.

7 Acknowledgement

This research is supported in part by the research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] Gianni Amati. Probabilistic models for information retrieval based on divergence from randomness. *PhD thesis, Department of Computing Science, University of Glasgow*, 2003.

[2] Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, 19(1):1–27, 2001.

[3] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996.

[4] J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. 1971.

[5] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[6] Alan F. Smeaton, Paul Over, and Wessel Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In Ajay Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[7] Chengxiang Zhai. Notes on the lemur tfidf model, http://lemurproject.org/lemur/tfidf.pdf. October 2001.