DCU at VideoClef 2008

Eamonn Newman and Gareth J.F. Jones
CDVP, School of Computing, Dublin City University
{eamonn.newman|gareth.jones}@computing.dcu.ie

Abstract

We describe a baseline system for the VideoCLEF Vid2RSS task. The system uses an unaltered off-the-shelf Information Retrieval system. ASR content is indexed using default stemming and stopping methods. The subject categories are populated by using the category label as a query on the collection, and assigning the retrieved items to that particular category. We describe the results of the system and provide some high-level analysis of its performance.

General Terms

ASR, IR, Classification

Keywords

Classification, Information Retrieval, Automatic Speech Recognition

1 Introduction

We implemented a system for the generation of topic-based RSS feeds of dual language audio-visual content for the Vid2RSS task [2]. Our system provides a baseline based on an Information Retrieval approach. We built a standard free text index using videos' ASR transcripts (and metadata) as the content. To populate a particular feed, the feed label (i.e., the category name) was used as a query to the search engine. The retrieved results were used as the component items of the feed. Some of the runs allowed an item to appear in one feed only, while others allowed items to appear in multiple feeds. These are described in full detail in Section 3.

2 System Description

We used the open source Lucene Search Engine technology [1] as the base technology for our system. Dutch-language content was stopped, stemmed and tokenised using Lucene's built-in Dutch analyser, `DutchAnalyzer`[1]. English-language content was stopped and tokenised by the Lucene default tokeniser, `StandardAnalyzer`[2]. The `StandardAnalyzer` does not perform any stemming of tokens.

For indexing the ASR transcripts, we parsed the documents and processed all `FreeTextAnnotation` elements. We did not make any use of the timestamp information available. For indexing the metadata documents (for Run 5), we indexed all `description` elements. All other metadata fields were discarded.

The feed categories were populated by using the label of each feed as a query. Feeds were populated in the order given in Table 1. The labels were arranged in order of most specific to

[1] org.apache.lucene.analysis.nl.DutchAnalyzer
[2] org.apache.lucene.analysis.standard.StandardAnalyzer

least specific. This meant that when an item was retrieved, it was placed into the most specific category possible. In Runs 1,3, and 5, the item was categorised only once. In Runs 2 and 4, it was placed in all categories for which it was retrieved.

Dutch	English
archeologie	archeology
architectuur	architecture
chemie	chemistry
dansen	dance
schilderijen	paintings
wetenschappelijk onderzoek	scientific research
beeldende kunst	visual arts
geschiedenis	history
film	film
muziek	music

Table 1: Category Label Order

Items were permitted to appear in multiple feeds in two of the runs (Runs 3 and 4), but were restricted to single appearances in the other runs. The restriction was imposed to improve the precision of the classification task, since labels such as "film" were very general and tended to capture most, if not all, of the items.

3 Run Configurations

In this section, we describe each of the separate runs which were submitted to the task.

1. **Dutch ASR transcripts:** In this run, we indexed the entire set of Dutch ASR transcripts (the `FreeTextAnnotation` elements). The index was queried with the labels in the order given above and the feeds were populated.

2. **English ASR transcripts:** This is identical to Run 1, using English ASR transcripts and translations of the category labels as queries.

3. **Dutch ASR with blind relevance feedback:** We ran the same queries as Run 1, but added an additional step of blind relevance feedback, to expand the query terms. Since this action reduces the precision of any query, we also relaxed the restriction on items appearing in only one feed. Queries were expanded by performing an initial query which consisted of just the category label. We take the first 10 retrieved documents and extract the 5 most frequently occuring terms in each. We process this set of 50 terms to remove any duplicates. The remaining terms are combined with the original query to form the expanded query.

4. **English ASR with blind relevance feedback:** This is identical in method to Run 3, but is the data now consists of the English ASR transcripts, rather than the Dutch.

5. **Dutch metadata:** We indexed the catalogue metadata from B&G which were supplied in the data sets. Specifically, we used the `description` elements from the metadata documents. Once again, the Dutch category term labels were used as queries, and the items were restricted to appear in one feed only.

4 Results

In Table 2 we present the retrieval scores attained by our system runs. A direct comparison of Runs 1 and 2 suggest that the Dutch transcripts were more useful in identifying the subject categories than the English ones. Indeed, the English transcripts had the poorest f-scores at both

metric	Run 1	Run 2	Run 3	Run 4	Run 5
micro-average precision	0.50	0.32	0.16	0.17	0.83
micro-average recall	0.35	0.21	0.91	0.72	0.18
f-score micro-average	0.41	0.25	0.28	0.28	0.29
macro-average precision	0.54	0.62	0.42	0.50	0.93
macro-average recall	0.55	0.38	0.90	0.70	0.28
f-score macro-average	0.54	0.47	0.58	0.59	0.43

Table 2: Vid2RSS Scores for Runs 1 to 5

micro and macro level. This is likely attributable to the fact that the majority of the dialogue was in Dutch and so contained less "noise" than the English counterparts. Processing of the ASR transcripts to identify the points at which the language changed would allow for the combination of transcripts (or the removal of erroneous segments) which would improve classification performance.

Runs 3 and 4 used relevance feedback to expand the queries and allowed items to be placed in multiple categories. As we can see, this relaxation resulted in a large drop in the micro-average precision scores of these systems; conversely, the micro-average recall is much higher in these runs. As items were placed in multiple categories, so the chances of an item being correctly classified was much greater, however the number of false positives also increased.

As can be seen from the results, Run 5 performed particularly well in terms of precision, and relatively well (when compared to our other runs) in terms of Recall. However, since this was on the metadata, and not on the ASR transcripts it cannot be directly compared to the others. The higher precision scores do suggest that there may be merit in combining the different data sets available.

One immediately obvious drawback with this system is that it is not possible to guarantee that all items will be classified. If an item is not retrieved for any of the queries, then it will not be placed in any of the category feeds. As it happens, this was not the case for any of the runs with this particular data set (probably due to the presence of highly generic labels such as "film" and "music")

Additionally, the number of terms added in the query expansion phase could be reduced. The maximum for this was 50, but elimination of duplicates meant that the size of the set was generally much smaller. Nevertheless, it seems that too many terms were added to the queries, and this is supported by the difference in micro-average precision between Runs 1 and 3 and Runs 2 and 4.

5 Conclusions

These notes discussed the baseline system implemented in DCU to address the VidCLEF challenge. We outlined the system architecture and how data was processed. The parameters for each run were given and the results of these were examined and analysed.

The results suggest that there is room for improvement in our system. The precision scores could be improved by finer-grained query expansion, which will be examined in the next set of experiments we carry out. Additionally, the performance on the English-language content could be improved by use of a stemming algorithm, such as Porter [3].

References

[1] Apache Software Foundation. Lucene: Java-based Indexing and Searching technology, http://lucene.apache.org/.

[2] Martha Larson, Eamonn Newman, and Gareth Jones. Overview of VideoCLEF 2008: Automatic generation of topic-based feeds for dual language audio-visual content. In *CLEF 2008 workshop notes, eds: Borri, F., Nardi, A. and Peters, C.,*, 2008.

[3] Martin Porter. An algorithm for suffix stripping. *Progam*, July 1980.