Evaluation of multimedia retrieval systems

Djoerd Hiemstra and Wessel Kraaij
University of Twente and TNO ICT

February 15, 2007

1 Introduction

In this chapter, we provide the tools and methodology for comparing the effectiveness of two or more multimedia retrieval systems in a meaningful way. Several aspects of multimedia retrieval systems can be evaluated without consulting the potential users or customers of the system, such as the query processing time (measured for instance in milliseconds per query) or the query throughput (measured for instance as the number of queries per second). In this chapter, however, we will focus on aspects of the system that influence the *effectiveness* of the retrieved results. In order to measure the effectiveness of search results, one must at some point consult the potential user of the system. For, what are the correct results for the query "black jaguar"? Cars, or cats? Ultimately, the user has to decide.

Doing an evaluation involving real people is not only a costly job, it is also difficult to control and therefore hard to replicate. For this reason, methods have been developed to design so-called test collections. Often, these test collections are created by consulting potential users, but once they are created they can be used to evaluate multimedia retrieval systems without the need to consult the users during further evaluations. If a test collection is available, a new retrieval method or system can be evaluated by comparing it to some well-established methods in a controlled experiment. Hull [15] mentions the following three ingredients of a controlled retrieval experiment.

1. A test collection consisting of 1) multimedia data, 2) a task and data needed for that task, for instance image search using example images as queries, and 3) ground thruth data or so-called relevance judgements (i.e., the correct answers);

2. One or more suitable evaluation measures that assign values to the effectiveness of the search;

3. A statistical methodology that determines whether the observed differences in performance between the methods investigated are statistically significant.

Multimedia retrieval systems typically combine many components and tools. For instance, for a video retrieval system, a component might be a large-vocabulary speech recognition system; other components might detect low level feature representations such as colour histograms and Gaussian mixtures; the system might have a component that performs shot detection and key frame selection based on the low-level features, to allow the user to search for video shots, etc., etc. We assume that a system component is an input-output device: feed in a video and get out some annotation of the video data. We also assume that components may be pipelined: some components will input the output of one or more other components to produce annotations. The quality of a system component will affect the quality of the system as a whole. So, there are two approaches to test the effectiveness of complex systems such as multimedia search systems [13]:

1. *glass box* evaluation, i.e., the systematic assessment of every component of a system, and

2. *black box* evaluation, i.e., testing the system as a whole.

For some of the components mentioned above, it is easy to define a clear task, and it is possible to come up with meaningful ground truth data. For instance, for the speech recognition subtask, we can ask a human annotator to define the ground truth speech transcript of the test data; we can run our large-vocabulary speech recognition system on the data; and we can compute the word-error rate (the percentage of words that was wrongly recognised). Similarly, for the shot detection task, we can ask a human annotator to mark the shot boundaries; we can run our shot detector; and report precision and recall of the shot detection component (see Section 3 and 5). For complex systems that are composed of such components, glass box evaluation is a good option. An advantage of a glass box evaluation procedure is that it easily pinpoints the parts of the systems that should be improved.

For other components it is less clear what the ground truth data should be. For instance, for a component that extracts colour histograms or Gaussian mixture models, it is impossible to say what constitutes a good histogram or mixture. The only way to evaluate a low-level feature extraction component, is to do a black box evaluation: Test the system as a whole with component A; then test a system as a whole with component B; and see if there is a significant difference. Even components for which we seem to be able to do a meaningful evaluation – such as the speech recognition and shot detection components mentioned above – we do not know upfront what the effect of the component's quality will be on the overall system quality. In general, a better speech recognition component will result in a better system, however, a 100% improvement in word error rate does in practice not result in a 100% improvement of precision and recall (see Section 3) of the overall system [12].

In this chapter, we will discuss the evaluation of multimedia retrieval systems by following the three ingredients of Hull: a test collection, an evaluation measure, and a statistical methodology. In Section 2, we will discuss some multimedia test collections and the retrieval tasks or subtask that can be evaluated with them. In Section 3, some well known evaluation measures are introduced. Section 4 briefly introduces significance testing. Finally, Section 5 will discuss the TRECVID collection and evaluation workshop in more depth.

2 Test collections and evaluation workshops

The scientific evaluation of multimedia search systems is a costly and laborious job. Big companies, for instance web search companies such as Yahoo and Google, might perform such evaluations themselves, but for smaller companies and research institutions a good option is often to take a test collection that is prepared by others. In order to share resources, test collections are often created by the collaborative effort of many research groups in so-called evaluation workshops of conferences. This approach was first taken in the Text Retrieval Conferences (TREC) [31]. The TREC-style evaluation approach is now taken by many smaller workshops that aim at the evaluation of multimedia search systems. An evaluation workshop typically follows a yearly cycle like the following: 1) The cycle starts with a call for participation in which an evaluation task and data is announced; 2) Groups that participate receive the data and possibly some ground truth training data; 3) Groups receive the test data (for instance queries) perform the test and send their results to the workshop organisers; 4) The organisers merge the results of all participants into one big pool create new ground truth data for those results; 5) The participants meet to discuss the evaluation results; 6) Participants publish the results as workshop proceedings. Often, an important side-effect of the evaluation workshop is the creation of a test collection that can be used later on without the need to create new ground truth data or relevance judgements.

For black-box evaluations, people are involved in the process of deciding whether a multimedia object is relevant for a certain query. The results of this process are the so-called *relevance judgements* or *relevance assessments*. The people involved in this process are usually called *assessors*. For glass-box evaluations, for instance an evaluation of a shot boundary detection algorithm, the process of deciding whether a sequence of frames contains a shot boundary involves people as well (for instance, for TRECVID 2004 almost 5,000 shot transitions in the test data were identified and classified by a single student that worked at the US National Institute of Standards and

Technology [17]). The results of this process are the so-called *ground truth data* or *reference data*.

In the following subsections, we briefly review a number of test collections that are publicly available, most of which are developed in evaluation workshops. The sections are meant to be illustrative. Workshops change their tasks regularly, workshops might stop their activities after a number of years and new workshops might emerge depending on trends in research.

2.1 The Corel set

The Corel test collection is a set of stock photographs, which is divided into subsets of images each relating to a specific theme (e.g. Arabian horses, sunsets, or English pub signs, see Figure 1). The collection is used to evaluate the effectiveness of content-based image retrieval systems in a large number of publications [3, 4, 5, 9, 16, 18, 30] and has become a de-facto standard in the field. Usually, Corel is used as follows: 1) From the test collection, take out one image and use it as a query-by-example; 2) Relevant images for the query are those that come from the same theme; and 3) Compute precision and recall for a number of queries.

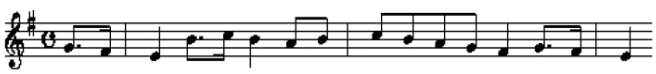
Theme: Arabic Horses

Figure 1: Example images from the Corel database

A problem with evaluation using Corel is that a single 'Corel set' does not exist. The data is sold commercially on separate thematic CDs, and different publications use different CDs and different selections of themes. Müller et al. [20] showed that evaluations using Corel are highly sensitive to the subsets and evaluation measures used. Another problem is that the data can be qualified as 'easy' because of the clear distinctions between themes and the high similarity within a theme because they usually come from one source (e.g. one professional photographer). Westerveld and De Vries [32] showed that good results on Corel do not guarantee good results in a more realistic setting, such as the TRECVID collection described below.

2.2 The MIREX workshops

The Music Information Retrieval Evaluation eXchange (MIREX) was organised for the first time in 2005 as a direct descendant of the Audio Description Contest organized in 2004 as part of the International Symposium on Music Information Retrieval (ISMIR). The MIREX test collections consist of data from record labels that allow to publish tracks from their artists' releases, for instance from internet record labels like Epitonic [10] and Magnatune [19] that feature music by independent artists.



Composer: Anonymous
Title: Roslin Castle
Signature: 000.109.446

Figure 2: Example symbolic music from the MIREX RISM A/II database

MIREX uses a glass box approach to the evaluation of music information retrieval by identifying various subtasks an musing retrieval system would have to perform and allow the participants to build components that perform those tasks. In 2005, the workshop identified the following tasks: Artist identification, Drum detection, Genre classification, Melody extraction, Onset detection, Tempo extraction, Key finding, Symbolic genre classification, Symbolic melodic similarity. The

tasks are performed on CD-quality audio except for the last two tasks which operate on symbolic music as shown in Figure 2 [8].

2.3 TRECVID: The TREC Video Retrieval workshops

An important evaluation workshop for this chapter is TRECVID: The TREC Video evaluation workshop. The workshop emerged as a special task of the Text Retrieval Conference (TREC) in 2001 and was later continued as an independent workshop collocated with TREC. The workshop has focused mostly on news videos, from US, Chinese and Arabic sources, such as CNN Headline News, and ABC World News Tonight. Figure 3 shows some TRECVID example data.

```
<VideoSegment id="shot88-16">
  <MediaTime>
   <MediaTimePoint>T00:02:48:16051F30000</MediaTimePoint>
   <MediaDuration>PT9S18288N30000F</MediaDuration>
  </MediaTime>
  <TextAnnotation confidence="0.581549">
   <FreeTextAnnotation>KNOW HOW I'LL A FAMILIAR SONG AND
DANCE SAY TO THE RUSSIAN PRESIDENT</FreeTextAnnotation>
  </TextAnnotation>
</VideoSegment>
```

Figure 3: Example data from TRECVID with an MPEG-7 annotation

TRECVID provides several black box evaluation tasks such as: Shot boundary detection, Story segmentation, and High-level feature extraction. The workshop series also provides a black-box evaluation framework: a Search task that may include the complete interactive session of a user with the system. We will describe the TRECVID tasks in depth in Section 5.

2.4 The multimedia tasks of the CLEF workshops

CLEF stands for Cross-Language Evaluation Forum, a workshop series mainly focussing on multilingual retrieval, i.e., retrieving data from a collection of documents in many languages, and cross-language retrieval, i.e., querying data using one natural language and retrieving documents in another language. Like TRECVID, CLEF started as a cross-language retrieval task in TREC in 1997. It became independent from TREC in 2000 and is currently run in Europe. Within CLEF, several image search tasks were organised starting in 2003. The goal of ImageCLEF is to investigate the effectiveness of combining text and image for retrieval [7]. Interestingly, a cross-language image search task is easier evaluated than a cross-language test search task: Whatever the language skills of the user, the relevance of an image is easily determined independently of the language of the caption or the language of the surrounding text.

Short title: Rev William Swan.
Location: Fife, Scotland
Description: Seated, 3/4 face studio portrait of a man.
Photographer: Thomas Rodger
Categories: [ministers][identified male] [dress - clerical]
Notes: ALB6-85-2 jf/ pcBIOG: Rev William Swan () ADD: Former owners of album: A Govan then J J? Lowson. Individuals and other subjects indicative of St Andrews provenance. By T. R. as identified by Karen A. Johnstone "Thomas Rodger 1832-1883. A biography and catalogue of selected works".

Figure 4: An example image from the CLEF St. Andrews database

There is a CLEF image databases consisting of historic photographs from the library at St. Andrews University [7], for which all images are accompanied by a caption consisting of several distinct fields, see Figure 4 for an example.

CLEF also provides an evaluation task for a medical image collection consisting of medical photographs, X-rays, CT-scans, MRIs, etc. provided by the Geneva University Hospital in the Casimage project. Casimage is the fusion of the words "case" and "Image". The goal of the

Chapter: Pathologie cardiaque
Diagnosis: Insuffisance mitrale
Clinical Presentation: Dyspné d'effort. Souffle systolique au foyer mitral
Description: Dilatation de l'oreillette gauche qui reste inscrite dans l'oreillette droite. Dilatation du ventricule gauche (arc inffieur gauche). Arc moyen gauche à double bosse: auricule gauche et tronc de l'artère pulmonaire dilatés
Commentary: L'insuffisance mitrale entrane une surcharge en volume de l'oreillette et du ventricule gauches responsable d'une dilatation de ces deux cavités. Le retentissement en amont explique la dilatation secondairedes cavités droites.

Figure 5: An example case from the CLEF Casimage database

project is to help the development of databases for teaching purposes. A database is a collection of cases, for instance diseases. Each case contains several images with textual descriptions [24]. CLEF provides black-box evaluations for these databases in the form a several search tasks.

2.5 The multimedia tasks of the INEX workshops

The Initiative for the Evaluation of XML Retrieval (INEX) aims at evaluating the retrieval performance of XML retrieval systems. The INEX multimedia track differs from other approaches in multimedia information retrieval, in the sense that it focuses on using the structure of the document to extract, relate and combine the relevance scores of different multimedia fragments.

```
<destination date-exported="2005-02-23">
  <name>Amsterdam</name>
   <general>
    <full-name>Amsterdam</full-name>
     <introduction>
      <mini>History, art, a head of beer and a roll-your-own.</mini>
        :
   </general>
   <images>
    <image filename="/images/BN13629-109.jpg">
     <caption>Neo-Renaissance facade of the <i>Stadsschouwburg</i>
       (City Theatre)</caption>
     <photographer>Martin Moos</photographer>
```

Figure 6: Example data from the INEX Lonely Planet database

INEX 2005 had a modest multimedia task that involved data from the Lonely Planet. A search query might for instance search for images by combining information on its photographer, the destination name, its caption text, but by looking for images similar to an example image [34].

3 Evaluation measures

The effectiveness of a system or component is often measured by the combination of *precision* and *recall*. Precision is defined by the fraction of the retrieved or detected objects that is actually relevant. Recall is defined by the fraction of the relevant objects that is actually retrieved.

$$\text{precision} = \frac{r}{n} \qquad r: \text{number of relevant documents retrieved}$$
$$n: \text{number of documents retrieved}$$
$$\text{recall} = \frac{r}{R} \qquad R: \text{total number of relevant documents}$$

Precision and recall are defined on sets of objects, not on ordered lists of objects. If the system ranks the objects in decreasing order of some score value, then the precision and recall measures should somehow be averaged over the number of objects retrieved. Several average precision and average recall measures have been suggested that model the behaviour of a user walking down a ranked list of objects. The idea is to give a number of evaluation measures for different types of users. At one end of the spectrum is the user that is satisfied with any relevant object, for

instance a user that searches a web page on last nights football results. At the other end of the spectrum is the user that is only satisfied with most or all of the relevant objects, for instance a lawyer searching for jurisprudence. We present four different evaluation measures that combine precision and recall in this section: precision at fixed levels of recall, precision at fixed points in the ranked list, the average precision over the ranks of relevant documents, and the F measure [14].

3.1 Precision at fixed recall levels

For this evaluation a number of fixed recall levels are chosen, for instance 10 levels: $\{0.1, 0.2, \cdots, 1.0\}$. The levels correspond to users that are satisfied if they find respectively 10 %, 20 %,$\cdots$, 100 % of the relevant documents. For each of these levels the corresponding precision is determined by averaging the precision on that level over the tests that are performed, for instance over a set of test queries. The resulting precision points are often visualised in a recall-precision graph. Figure 7 shows an example. The graph shows the typical behaviour of information retrieval systems. Increasing the recall of a search implies decreasing the precision of the search. Or, by walking down a ranked list in search for more relevant documents, the chance to encounter irrelevant documents will grow faster than the chance to encounter relevant documents.

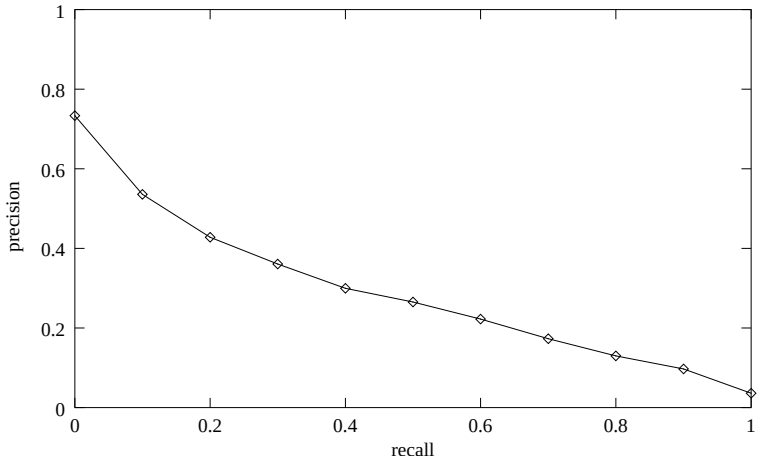


Figure 7: Example recall-precision graph

In practice, the levels of recall might not correspond with natural recall levels. For instance, if the total number of relevant documents R is 3, then the natural recall levels are 0.33, 0.67 and 1.0. Other recall levels are determined by using interpolation. A simple but often used interpolation method determines the precision at recall level l by the maximum precision at all points larger than l. For example, if the three relevant documents were retrieved at rank 4, 9 and 20, then the precision at recall points $0.0, \cdots, 0.3$ is 0.25, at recall points 0.4, 0.5 and 0.6 the precision is 0.22 and at $0.7, \cdots, 1.0$ the precision is 0.15 [14]. Interpolation might also be used to determine the precision at recall 0.0, resulting in a total of 11 recall levels. Sometimes one average measure, the so-called 11 points interpolated average precision, is calculated by averaging the average precision values over the 11 recall points.

3.2 Precision at fixed points in the ranked list

Recall is not necessarily a good measure of user equivalence. For instance if one query has 20 relevant documents while another has 200. A recall of 50 % would be a reasonable goal in the first case, but unmanageable for most users in the second case [15]. A more user oriented method would simply choose a number of fixed points in the ranked list, for instance 9 points at: 5, 10, 15, 20, 30, 100, 200, 500 and 1000 documents retrieved. These points correspond with users that are willing to

read 5, 10, 15, etc. documents of a search. For each of these points in the ranked list, the precision is determined by averaging the precision on that level over the queries. Similarly, the average recall might be computed for each of the points in the ranked list. A potential problem with these measures however is that, although precision and recall theoretically range between 0 and 1, they are often restricted to a small fraction of the range for many cut-off points. For instance, if the total number of relevant documents $R = 3$, then the precision at 10 will be 0.3 at maximum. One point of special interest from this perspective is the precision at R documents retrieved. At this point the average precision and average recall do range between 0 and 1. Furthermore, precision and recall are by definition equal at this point. The R-precision value is the precision at each (different) R averaged over the queries [14].

3.3 Mean average precision

The average precision measure is a single value that is determined for each request and then averaged over the requests. The measure corresponds with a user that walks down a ranked list of documents that will only stop after he / she has found a certain number of relevant documents. The measure is the average of the precision calculated at the rank of each relevant document retrieved. Relevant documents that are not retrieved are assigned a precision value of zero. For the example above where the three relevant documents are retrieved at ranks 4, 9 and 20, the average precision would be computed as $(0.25 + 0.22 + 0.15)/3 = 0.21$. This measure has the advantages that it does not need the interpolation method and that it uses the full range between 0 and 1 [14].

3.4 Combining precision and recall: the F measure

For many glass box evaluations, there is no ranked list that needs to be considered. For instance, a tool that detects video shot boundaries (see Section 5.1) does not really rank anything: It either detects a shot boundary or not. In such cases, we need to choose between two systems based on the raw precision and recall measures. Suppose that on a shot boundary detection task one system achieves a precision of 0.8 and a recall of 0.95, and another system achieves a precision of 0.99, but a recall of only 0.7. What system would we prefer? Obviously that depends on our preferences: do we want a system that detects shot boundaries accurately or thoroughly? If we are equally interested in precision *and* recall, we might use the following measure:

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{1}$$

The F-measure combines precision and recall with an equal weight. In the example above, the first system achieves $F = 0.87$, whereas the second system achieves $F = 0.82$. So, we better choose the first system if we are equally interested in precision and recall.

3.5 Problems with measuring recall

The computation of recall is a well-known problem in retrieval system evaluation, because it involves the manual assessment or estimation of the total number of relevant items in the database for each query. Assessment of each document is too costly for large collections and estimation by assessing a sample with sufficient reliability would still require large samples [28]. A common way to get around this problem is to use the *pooling method* which is applied in TREC. This method computes *relative* recall values instead of *absolute* recall. It is assumed that if we have a 'pool' of diverse retrieval systems, the probability that a relevant document will be retrieved by one of the systems is high. So a merged list of document rankings is assumed to contain most relevant documents. The pool assumption is actually a bit more precise: we assume that most relevant documents are contained in a pool consisting of the merged top D documents of several different high quality retrieval systems. Here D is the *pool depth*, i.e., the number of retrieved items taken from the top of a retrieval run. At TREC usually a pool depth of 100 documents has been applied.

So, for each query, the top 100 documents from the submissions that contribute to the pool are merged in a set of which a list of unique document identifiers is extracted. These documents are subjected to a (manual) relevance assessment procedure.

In the previous sections we already mentioned that systems are often evaluated on existing test collections. There are two potential problems with "re-using" a test collection: (i) it takes more discipline to perform a really blind experiment and extra care not to tune on the data; (ii) post-hoc runs are unjudged runs by definition. An unjudged run is a run that did not contribute to the pool. For judged runs we know that at least the top 100 (the most common pool depth) is judged. For unjudged runs, this will not be the case. The percentage of judged documents (the judged fraction) will be lower. However, presenting results of unjudged runs is very common. Even at TREC not every run is judged. Participants can submit runs and because of the limited capacity and budget, the pool is based on a selection of the submitted runs, usually one run per participating site. For judged runs, the number of judged documents in the top 100 is exactly 100, for unjudged runs this number is lower. That means that the calculated performance measures are more reliable for judged runs. The difference in reliability between judged and unjudged runs has been studied by Zobel [33] as follows: recompute the average precision of every run that contributed to the pool based on a pool without the judged documents that were uniquely contributed by the very same run. Finally, compute the averages of the average differences or improvements in performance over the runs. He reported an average improvement of 0.5% over 61 runs with a maximum of 3.5% for the TREC-5 collection. The fact whether a run is judged or not thus seems to play a minor role in the TREC-5 dataset.

4 Significance tests

Simply citing percentages improvements of one method over another is helpful, but it does not tell if the improvements were in fact due to differences of the two methods. Instead, differences between two methods might simply be due to random variation in the performance, that is, the difference might occur by chance even if the two methods perform equally well. To make significance testing of the differences applicable, a reasonable amount of queries is needed. When evaluation measures are averaged over a number of queries, one can obtain an estimate of the error associated with the measure [15].

Often, a technique called *cross-validation* is used to further prevent biased evaluation results. The data and ground truth data is splitted in two disjoint sets: the *training set* and the *test set*. The training set is used for development and tuning of the system and, if applicable, to train learning algorithms like classifiers. Then, the test set is used to measure the effectiveness of the system. These three steps (dividing the data, training and testing) might be repeated for different sizes of the sets and different random selections of training set and test set for each size.

4.1 Assumptions of significance tests

Significance tests are designed to disprove the null hypothesis H_0. For retrieval experiments, the null hypothesis will be that there is no difference between method A and method B. The idea is to show that, given the data, the null hypothesis is indefensible, because it leads to an implausible low probability. Rejecting H_0 implies accepting the alternative hypothesis H_1. The alternative hypothesis for the retrieval experiments will be that either method A consistently outperforms method B, or method B consistently outperforms method A.

A test statistic is a function of the data. It should have the following two properties. Firstly, it should behave differently under H_0 than under H_1. Secondly, it should be possible to calculate its probability distribution under H_0. For information retrieval, there is usually much more variation in the performance per query than in the performance per system. Therefore, the test statistics used are paired tests which are based on the performance differences between two systems for each query. The methods assume that the performance differences consist of a mean difference μ and

an error ε_i for each query i, where the errors are independent. The null hypothesis is that $\mu = 0$. The following three paired tests have been used in the Smart retrieval experiments [25, page 171].

the paired t-test assumes that errors are normally distributed. Under H_0, the distribution is Student's t with $\#queries - 1$ degrees of freedom.

the paired Wilcoxon's signed ranks test is a non-parametric test that assumes that errors come from a continuous distribution that is symmetric around 0. The statistic uses the ranks of the absolute differences instead of the differences themselves.

the paired sign test is a non-parametric test that only uses the sign of the differences between method A and B for each query. The test statistic is the number of times that the least frequent sign occurs. It assumes equal probability of positive and negative errors. Under H_0, the distribution is binomial.

So, in order to use the t-test the errors must be normally distributed, and in order to use Wilcoxon's test the errors have to be continuous. However, precision and recall are discrete and bounded and therefore neither normally distributed nor continuous. Still, the average of a reasonable number of discrete measures, like the average precision measure presented in section 3.3, might behave similar to continuous measures and approximate the normal distribution quite well. Before the tests can be applied, the researcher has to make a qualitative judgement of the data, to check if indeed the normality assumption is reasonable [15]. If not, the sign test can be used as an alternative. Some researchers, for instance Van Rijsbergen [29], argue that only the sign test can be considered valid for information retrieval experiments.

4.2 Example: performing a significance test

As an example consider the following experiment. We compare the effectiveness of two content-based image retrieval systems A and B that, given an example query image, return a ranked lists of images from a standard benchmark test collection. For each system, we run 50 queries from the benchmark. Using the benchmark's relevance judgements, we compute the average precision for each query on both systems. Suppose the mean average precision is 0.208 for system A and 0.241 for system B. Are these differences significant? To test this, we might use a two-tailed pair-wise sign test and only report differences at a 1 % level as significant, that is, the probability that differences are contributed to chance should be less than 1 %.

To apply the test, we determine for each query the sign of the difference between the average precision of both systems: -1 if system A outperforms system B on the query, and 1 if system B outperforms system A. Let's assume that it turns out that for 16 queries system A produced the highest average precision, and for 34 queries system B produced the highest average precision. Under H_0, the probability of observing these numbers can be computed by the binomial distribution. However, it would be incorrect to only calculate the probability of getting exactly 16 successes for system A. Instead, we should calculate the probability of getting a deviation from the null hypothesis as large as, or larger than, the observed result. The probability of getting k or less successes in n trials with probability p is given by the cumulative binomial distribution function F:

$$F(k; n, p) = \sum_{j=1}^{k} \frac{n!}{j!(n-j)!} p^j (1-p)^{n-j} \tag{2}$$

In this case, $F(16; 50, 0.5) = 0.00767$. So, the probability of observing 16 or less successes for system A is 0.00767, which is less than 1 %. But we are not done yet! The above calculation gives the total probability of getting 16 or less successes for system A. However, the alternative hypothesis H_1 states that the probability of success for system A is not equal to the probability of success of system B. If there had been 34 successes for system A, then that would have been an equally extreme deviation from the expected number of successes under the null hypothesis. To account for both tails of the probability distribution – hence the name *two-tailed* test – the

probability has to be multiplied by 2, resulting in a probability of 0.0153. Therefore, we cannot reject the null hypothesis. Despite the substantial difference between the mean average precision of both systems, this might still be contributed to chance. The two-tailed pair-wise sign test was unable to detect a significant difference at the 1 % level.

5 A case study: TRECVID

In 2004, there were four main tasks in TRECVID: shot boundary detection; story segmentation; high-level feature extraction; and search. We will describe each of these four tasks below.

5.1 Shot boundary detection

When you see a film in the theatre or on television, you see moving images, but actually the film consists of still pictures, called *frames*. If enough different frames are projected within each second, typically 25 or 30 per second, the human brain smears them together and we get the illusion of motion or change. Because there are many nearly identical frames each second, it does not make much sense to index each single frame when building a video search system. The system would typically index video at some higher granularity, for instance by grouping frames together into *shots*. Traditionally, a shot marks the complete sequence of frames starting from the moment that the camera was switched on, and stopping when it was switched off again, or a subsequence of that selected by the movie editor. Shot detection is a crucial step in many video search systems. Shot retrieval is for instance useful in a scenario where a user has the desire to re-use video material, or in a scenario where a user wants very precise points into the video stream.

A shot detector is a tool that inputs digital video and outputs the shot boundaries, i.e., the positions in the video stream that contain the transitions from one shot to the next shot. Usually, at least two types of shot transitions are distinguished: *hard cuts* and *soft cuts*. A hard cut is an abrupt change from one shot to the next shot: One frame belongs to the first shot, and the next frame to the second shot. Hard cuts are relatively easy to detect because there is usually a relatively large difference between the two frames. A soft cut, or *gradual transition*, uses several frames to establish the cut, that is, there is a sequence of frames that belongs to both the first shot and the second shot. These are much harder to detect, because there are no obvious points in the video with a big difference between consecutive frames that mark the shot boundary. Gradual transitions are sometimes further classified as *dissolves*, *fades* and *wipes*.

The TRECVID shot boundary task is defined as follows: identify the shot boundaries with their location and type (hard or soft) in the given video clips. The performance of a system is measured by precision and recall of detected shot boundaries, where the detection criteria require only a single frame overlap between the submitted transitions and the reference transition. This is done to make the detection independent of the accuracy of the detected boundaries. For the purposes of detection, a hard cut transition is treated as if it includes both the frame before the cut and after the cut, making the length of a hard cut effectively two frames instead of zero frames.

The TRECVID 2004 test data consists of 618,409 frames in total containing 4,806 shot transitions of which about 58 % are hard cuts and 42 % are soft cuts, mostly dissolves. The best performing systems perform very well on hard cuts: around 95 % precision and 95 % recall, but considerably worse on soft cuts, which perform around 80 % precision and 80 % recall [17].

5.2 Story segmentation

A digital video retrieval system with shots as the basic retrieval unit might not be desirable in situations in which users are searching an archive of video assets that consist of a compilation of different topics, such as news shows and news magazines. In such a case, retrieval of a complete topical unit is usually preferred over retrieval of shots. However, reverse-engineering the structure of e.g. a television news show in a general fashion is not always straightforward, since news shows have widely differing formats. The segmentation of a news show into its constituting news items

has been studied under the names of *story boundary detection* and *story segmentation*. Story segmentation of multimedia data can potentially exploit visual, audio and textual cues present in the data. Story boundaries often but not always occur at shot boundaries, since an anchor person can conclude a story and introduce the subsequent story in a single shot. A news item often spans multiple shots, starting with an anchor person, switching e.g. to a reporter on-site or a split screen interview.

Story segmentation of digital video was evaluated at TRECVID in 2003 and 2004 based on a collection of ABC/CNN news shows. These news shows consist of a series of news items interspersed with publicity items. The story segmentation task was defined as follows: given the story boundary test collection, identify the story boundaries with their location (time) in the given video clip(s). The definition of the story segmentation task was based on manual story boundary annotations made by Linguistic Data Consortium (LDC) for the Topic Detection and Tracking (TDT) project [11] and thus LDC's definition of a story was used in the task. A news story was defined as a segment of a news broadcast with a coherent news focus which contains at least two independent, declarative clauses. Other coherent non-news segments were labelled as *miscellaneous*, merged together when adjacent, and annotated as one single story. These non-news stories cover a mixture of footage: commercials, lead-ins and reporter chit-chat.

With the TRECVID 2003 and 2004 story segmentation task, the goal was to show how video information can enhance or completely replace existing story segmentation algorithms based on text, in this case Automatic Speech Recognition (ASR) transcripts and/or Closed Captions (CC). In order to concentrate on this goal there were several required experiments (called *runs* in TRECVID) for participants in this task:

- Video + Audio (no ASR/CC)
- Video + Audio + ASR/CC
- ASR/CC (no Video + Audio)

Additional optional runs using other ASR and/or CC-based transcripts were also allowed to be submitted. Since story boundaries are rather abrupt changes of focus, story boundary evaluation was modelled on the evaluation of shot boundaries (the hard cuts, not the gradual boundaries). A story boundary was expressed as a time offset with respect to the start of the video file in seconds, accurate to nearest hundredth of a second. Each reference boundary was expanded with a fuzziness factor of five seconds in each direction, resulting in an evaluation interval of 10 seconds. A reference boundary was detected when one or more computed story boundaries lay within its evaluation interval. If a computed boundary did not fall in the evaluation interval of a reference boundary, it was considered a false alarm. In addition, the F-measure (see Section 3.4) was used to compare performance across conditions and across systems.

5.3 High-level feature extraction

One way to bridge the semantic gap between video content and textual representations is to annotate video footage with *high level features*. The assumption is (and evidence is accumulating [1, 27]) that features describing generic concepts such as **indoor/outdoor** or **people** could help search and navigation. It is easy to understand the plausibility of this assumption, when a collection of video assets is annotated (at the shot level) with a lexicon of (high level) features, these features can be used to constrain the search space by conditioning the result set to include or exclude a certain feature. High level features can thus complement search in speech recognition transcripts/closed captions. A sufficiently rich and structured feature lexicon could strongly enhance the possibilities for faceted navigation, i.e., navigation where a user interactively selects features from different concept classes.

In TRECVID, the high level feature detection task is modelled as a ranking task. Systems have to assign to each shot (taken from a standard collection) the probability that a certain feature is present. The top 2000 shots for each feature are submitted for manual evaluation, based on the

pooling principle described in Section 3.5. This approach has the advantage that the number of manual judgements that has to be done is limited. The presence/absence of a feature is taken to be a binary property, if a feature holds for just one frame in a shot, it is assumed the feature is present for the whole shot. For each feature, the average precision is computed as described in Section 3.3 using the standard TREC evaluation package [6]. Since the test features in TRECVID cannot be thought of as a random draw from the space of possible features (the features represent fixed rather than random factors), average precision is reported per feature and no *mean* average precision is calculated.

5.4 Video search

The shot detection, story segmentation and high-level feature extraction tasks discussed up till now are glass box evaluations. The purpose of the *video search* task however, is to evaluate the system as a whole, i.e., to do a black-box evaluation of the system. The task is as follows: Given the search test collection and a multimedia statement of a user's information need (called *topic* in TRECVID), return a ranked list of at most 1000 shots from the test collection that best satisfy the need. A topic consists of a textual description of the user's need, for instance: "Find shots of one or more buildings with flood waters around it/them." and possibly one or more example frames or images. Researchers might use the topics to automatically generate queries from, for instance the textual description might be used to generate a text query which is run on automatic speech recognition transcripts and each example frame might be used directly as a query-by-example, i.e., it might be used to retrieve shots that contain frames that are in some sense similar to the examples. However, the topics are intended to be used by human searchers who either manually formulate their queries, or interactively use the system to formulate and reformulate the queries. These two search tasks are shown in Figure 8 which was taken from the TRECVID guidelines [22].

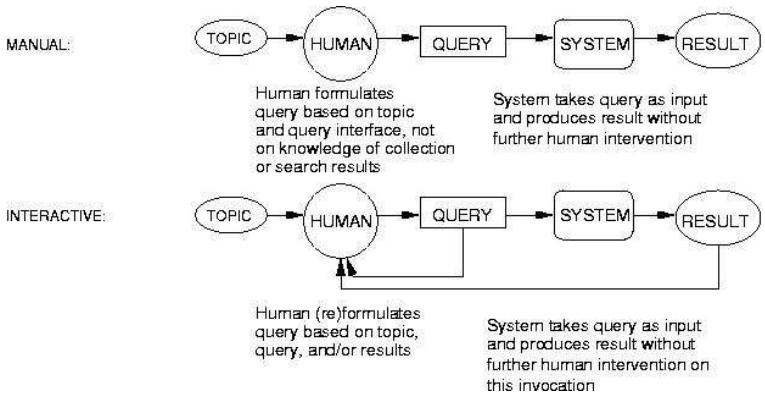


Figure 8: Two video search tasks: manual and interactive

The searcher should have no experience of the topics beyond the general world knowledge of an educated adult. The maximum total elapsed time limit for each topic (from the time the searcher sees the topic until the time the final result set for that topic is returned) in an interactive search run is 15 minutes. For manual runs the manual effort (topic to query translation) for any given topic is limited to 15 minutes as well. To make the search results of different systems better comparable, all systems use exactly the same shot boundaries provided by TRECVID, the so-called *common shot boundary reference*. This way, the shots act as pre-defined units of retrieval just as in ordinary (text) document retrieval. Retrieval methods are evaluated by comparing their mean average precision or by comparing the precision at several recall points.

6 Summary

Evaluation is a crucial element of multimedia retrieval research, since it is important to validate whether a certain idea or theoretical model is effective (and efficient) in practice. Test collections play an important role for the advancement of the field as a whole because they provide common reference points for measuring progress. Several rigorous evaluation methods exist for the performance measurement of fully automatic systems. Evaluation of systems that include user interaction is much more complicated, efficient evaluation methodology schemas are still under development for this area.

7 Further reading

Retrieval system evaluation methodology is described is many books on information retrieval. For instance, Keith van Rijsbergen [29] extensively discusses evaluation methodology. More up-to-date is the book by Ricardo Baeza-Yates and Berthier Ribeiro-Neto [2]. David Hull [15] wrote a well-cited paper on significance testing, discussing amongst others interactive search scenarios that include relevance feedback. Stephen Robertson [23] wrote an excellent evaluation tutorial for the European Summer School on Information Retrieval, ESSIR. John Smith [26] as well as Henning Müller et al. [21] wrote interesting notes on the specific issues for image and video retrieval.

References

[1] A. Amir, J. Argillandery, M. Campbellz, A. Hauboldz, G. Iyengar, S. Ebadollahiz, F. Kangz, M.R. Naphadez, A. Natsevz, J.R. Smithz, J. Tesicz, and T. Volkmer. IBM Research TRECVID-2005 Video Retrieval System. In *Proceedings of the TRECVID 2005 workshop*, 2005.

[2] R. Baeza-Yates and B. Ribeiro-Neto, editors. *Modern Information Retrieval*. Addison-Wesley, 1999.

[3] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[4] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using em and its application to content-based image retrieval. In *Proceedings of the sixth International Conference on Computer Vision*, 1998.

[5] D.M. Blei and M.I. Jordan. Modeling annotated data. In *Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2003.

[6] C. Buckley. trec_eval: TREC evaluation software. In *Provided to participants of the Text Retrieval Conferences (TREC)*, 2006. http://trec.nist.gov/trec_eval/.

[7] P. Clough, H. Müller, and M. Sanderson. The CLEF cross-language image retrieval track (ImageCLEF) 2004. In *Proceedings of the fifth Workshop of the Cross Language Evaluation Forum (CLEF)*, Lecture Notes in Computer Science (LNCS). Springer, 2005.

[8] J.S. Downie, K. West, A. Ehmann, and E. Vincent. The 2005 music information retrieval evaluation exchange (MIREX 2005): preliminary overview. In *Proceedings of the Lnternational Conference on Music Information Retrieval*, 2005.

[9] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the seventh European Conference on Computer Vision*, pages 97–112, 2002.

[10] Epitonic. http://www.epitonic.com.

[11] J.G. Fiscus, G. Doddington, J.S. Garofolo, and A. Martin. NIST's 1998 Topic Detection and Tracking Evaluation (TDT2). In *Proceedings of the DARPA Broadcast News Workshop*, 1999.

[12] J. Garfolo, C. Auzanne, and E.M. Voorhees. The TREC SDR track: A success story. In *Proceedings of the eighth Text Retrieval Conference (TREC)*, pages 107–129, 2000.

[13] E.E.W. Group. Evaluation of natural language processing systems. Technical report, ISSCO, 1996.

[14] D.K. Harman. Appendix b: Common evaluation measures. In *Proceedings of the 13th Text Retrieval Conference (TREC)*, 2005.

[15] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th ACM Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 329–338, 1993.

[16] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2003.

[17] W. Kraaij, A.F. Smeaton, P. Over, and J. Arlandis. Trecvid - an introduction. In *Proceedings of TRECVID 2004*, 2004. http://www-nlpir.nist.gov/projects/trecvid/.

[18] J. Li and J.Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), 2003.

[19] Magnatune. http://magnatune.com.

[20] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about Corel – evaluation in image retrieval. In *Proceedings of The Challenge of Image and Video Retrieval (CIVR)*, 2002.

[21] H. Müller, W. Müller, D. McG.Squire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters*, 22:593–601, 2001.

[22] P. Over, A. Smeaton, and W. Kraaij. Guidelines for the TRECVID 2004 evaluation. 2004. http://www-nlpir.nist.gov/projects/tv2004/tv2004.html.

[23] S.E. Robertson. Evaluation in information retrieval. In M. Agosti, F. Crestani, and G. Pasi, editors, *European Summer School on Information Retrieval (ESSIR)*, number 1980 in Lecture Notes in Computer Science, pages 81–92. Springer-Verlag, 2000.

[24] A. Rosset, O. Ratib, A. Geissbuhler, and J.P. Vallé. Integration of a multimedia teaching and reference database in a pacs environment. *RadioGraphics*, 22:1567–1577, 2002.

[25] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[26] J. Smith. Image retrieval evaluation. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL)*, 1998.

[27] C.G.M. Snoek, J.C. van Gemert, J.M. Geusebroek, B. Huurnink, D.C. Koelma, G.P. Nguyen, O. de Rooij, F.J. Seinstra, A.W.M. Smeulders, C.J. Veenman, and M. Worring. The MediaMill TRECVID 2005 Semantic Video Search Engine. In *Proceedings of the 2005 TRECVID workshop*, 2005.

[28] J.M. Tague. The pragmatics of information retrieval experimentation. In K. Sparck-Jones, editor, *Information Retrieval Experiment*, pages 59–102. Butterworths, 1981.

[29] C.J. van Rijsbergen. *Information Retrieval, second edition.* Butterworths, 1979. http://www.dcs.gla.ac.uk/Keith/Preface.html.

[30] N. Vasconcelos and A. Lippman. A probabilistic architecture for content-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 216–221, 2000.

[31] E.M. Voorhees and D. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

[32] T. Westerveld and A.P. de Vries. Experimental evaluation of a generative probabilistic image retrieval model on 'easy' data. In *Proceedings of the Multimedia Information Retrieval Workshop*, 2003.

[33] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 307–314, 1998.

[34] R. van Zwol, G. Kazai, and M. Lalmas. INEX 2005 Multimedia Track. In *Advances in XML Information Retrieval and Evaluation: 4th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX)*, Lecture Notes in Computer Science 3977, Springer, pages 497–510, 2006.