

Image annotation using clickthrough data

Theodora Tsirikla¹, Christos Diou^{2,3}, Arjen P. de Vries^{1,4}, Anastasios Delopoulos^{2,3}

¹ CWI, Amsterdam, The Netherlands

² Multimedia Understanding Group, ECE Dept., Aristotle University of Thessaloniki, Greece

³ Informatics and Telematics Institute, Centre for Research and Technology Hellas

⁴ Delft University of Technology, Delft, The Netherlands

Theodora.Tsirikla@cwi.nl, diou@mug.ee.auth.gr, arjen@acm.org, adelo@eng.auth.gr

ABSTRACT

Automatic image annotation using supervised learning is performed by concept classifiers trained on labelled example images. This work proposes the use of clickthrough data collected from search logs as a source for the automatic generation of concept training data, thus avoiding the expensive manual annotation effort. We investigate and evaluate this approach using a collection of 97,628 photographic images. The results indicate that the contribution of search log based training data is positive; in particular, the combination of manual and automatically generated training data outperforms the use of manual data alone. It is therefore possible to use clickthrough data to perform large-scale image annotation with little manual annotation effort or, depending on performance, using only the automatically generated training data. The datasets used as well as an extensive presentation of the experimental results can be accessed at <http://olympus.ee.auth.gr/~diou/civr2009/>.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing; H.3.1 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

Image annotation, concepts, supervised learning, search logs, clickthrough data, collective knowledge, implicit feedback

1. INTRODUCTION

The application of supervised machine learning approaches in the automatic annotation of images and videos with semantic concepts requires the availability of labelled samples to be used as training data. Such annotated samples

are typically generated manually, a laborious and expensive endeavour. Even though collaborative large-scale annotation efforts have been organised, e.g., in the context of the TRECVID evaluation benchmark [16] or under the guise of the Web-based ESP game [21], the bottleneck still remains, given, in particular, the large number of semantic concepts estimated to be desirable in order to achieve higher retrieval effectiveness than the current state-of-the-art [6]. The situation is further exacerbated by the poor generalisation of concept classifiers to domains other than their training domain [23]; this implies that for achieving effective annotation, individual content owners need to carry out their own manual annotation exercise, a continual task for the many collections that keep expanding over time with new data.

To compensate for the high cost in manually labelling training samples, research has recently moved towards the use of alternative data sources that are automatically acquired from the Web in order to be used for training concept classifiers [22, 13, 4, 19]. Such data sources include user-generated multimedia content annotated with user-defined tags (e.g., YouTube and Flickr) [13, 4, 19], as well as images and videos annotated with keywords automatically extracted from the text that surrounds them in the Web pages they are embedded in [22]. In this paradigm shift, Web communities unknowingly share in the generation of large amounts of labelled data.

The work presented in this paper is also concerned with the automatic generation of annotated training samples for building concept classifiers. Focussing on the specific case of image annotation, it proposes and investigates the use of a different (and thus far untapped) source for acquiring such examples: the *clickthrough data* logged by retrieval systems. These data consist of the queries submitted by the users of such systems, together with the images in the retrieval results that these users selected to click on in response to their queries. This information can be viewed as a type of users' *implicit feedback* [12] that provides a "weak" indication of the relevance of the image to the query for which it was clicked on [5]. We refine the notion of relevance in this assumption by considering that the queries for which an image was clicked provide in essence a "weak" description (or *annotation*) of the image's visual content. Our aim, therefore, is to investigate whether images with such search log-based annotations can serve as labelled samples in a supervised machine learning framework for training effective concept classifiers.

The primary advantage of using samples annotated either through the use of clickthrough data or by user-provided

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIVR '09, July 8-10, 2009 Santorini, Greece

Copyright 200X ACM 978-1-60558-480-5/09/07 ...\$5.00.

tags is that such data can be gathered or acquired in large quantities without any major effort on the part of content owners (even though search logs are not easily made publicly available due to privacy concerns). In addition, for the particular case of clickthrough data, no explicit user intervention is required, since these are gathered unobtrusively in search logs during the users’ search-related interactions. Furthermore, most content owners are able to collect their own search logs and therefore produce training data (and associated classifiers) that are adapted to their collections, rather than having to rely on the use of external tagged sources and deal with cross-domain applicability issues.

On the other hand, the major shortcomings are that automatically acquired labelled data are sparse (they only cover the part of the collection that has been previously accessed), and potentially noisy. Manual annotations are reliable and based on clear visual criteria pertaining to the samples’ visual content, whereas tags and logged queries tend to describe not only the visual content, but also the context of multimedia resources. This has been recently illustrated in an analysis showing that Flickr’s users annotate their photos with respect to both their content and their context by using a wide spectrum of semantic tags [17]. Nevertheless, the use of large amounts of “noisily labelled” data might be the key in dealing with this quality gap. In particular, clickthrough data (and also tags assigned in a collaborative manner) could be considered as having further noise reduction properties, given that they encode the collective knowledge of multiple past users, rather than the subjective assessment (or tag assignment) of a single person.

The usefulness of such clickthrough data in the particular application of automatic image annotation is examined in the remainder of this paper, which is organised as follows. Section 2 discusses related work on the use of (i) data sources other than manually labelled samples as training data in the annotation of multimedia content, and (ii) search logs in multimedia retrieval applications. Section 3 describes our approach, while Sections 4 and 5 present the set up and the results of our experiments. Section 6 concludes this paper by summarising our main contributions and findings.

2. RELATED WORK

The approach of using publicly available tagged resources as labelled samples for training concept classifiers has been examined by a number of participants (e.g., [13], [4], and [19]) in the most recent TREC Video Evaluation Workshop (TRECVID 2008). Such tagged resources (either images from sites such as Flickr [13, 4] or videos from YouTube [19]) were downloaded through these sites’ search services in response to text queries corresponding to the concept name [4] or to manually selected additional keywords [13, 19]. To reduce the potential noise, researchers also restricted the initial search to (YouTube) categories deemed relevant to the concept in question [19], or eliminated the resources with low visual similarity to the manually annotated TRECVID data [4]. The main observation in these studies was that classifiers trained on the manually annotated data outperformed those trained on the automatically acquired labelled samples, with the latter though working well for some concepts. Overall, much further research is needed in order to reach reliable conclusions on the usefulness of such data sources as training data, particularly for the complex task of video annotation.

The Information Retrieval (IR) field has exploited click-

through data in a number of different applications, e.g., for generating surrogate document representations [15], for query suggestion [2], and as training samples for learning retrieval functions [10]. In multimedia retrieval applications, their use has been more limited, most probably due to the lack of publicly available search logs from multimedia search engines. A study that did investigate their use in a multimedia setting employed search logs provided by a general purpose Web-based image search engine in order to rank images with respect to a given textual query [5]. This study viewed the clickthrough data as a bipartite graph, with one set of vertices corresponding to queries and the other to images, and an edge denoting that an image has been clicked for a query. Then, given a query, a Markov random walk model was applied to this *clickgraph* in order to probabilistically rank the images. To the best of our knowledge though, there has been no previous work on incorporating the clickthrough data into the concept learning process.

3. OUR APPROACH

This section describes our approach for selecting ‘training images’ based on clickthrough data; such images can then be employed as labelled samples for training concept classifiers in automatic image annotation.

3.1 Problem definition

A *concept* c corresponds to a clearly defined, non ambiguous entity and is represented by a set $\{N_c, K_c, D_c\}$, where N_c is the concept’s short *name*, K_c are *keywords* that are conceptually related to c , and D_c is a free-text, short *description* of c . An example is the concept with $N_c = \text{traffic}$, $K_c = \{\text{traffic, traffic jam, cars, road, highway}\}$ and description $D_c = \text{“Image showing a high density of vehicles on a road or highway”}$.

Given an image collection \mathbf{I} , our aim is to apply a method m that automatically generates for each concept c a training set $\mathbf{T}_{c,m}$ to be used in a supervised machine learning setting. To this end, method m needs to find a set $\mathbf{I}_{c,m}$ of images that contain the concept c (positive examples), as well a set $\mathbf{I}_{\bar{c},m}$ (disjoint to $\mathbf{I}_{c,m}$) that consists of images that do not contain c (negative examples). This work investigates methods that are based on the clickthrough data collected in search logs to produce the set $\mathbf{I}_{c,m}$. The generation of $\mathbf{I}_{\bar{c},m}$ is based on random selection.

3.2 Search log based positive sample selection

The simplest method for selecting positive samples for a concept c based on search log data is to consider the images that have been clicked for queries that *exactly match* the concept’s name N_c ; this constitutes method m denoted as *exact*. Clickthrough data though are sparse [5], since (i) images that are relevant may not have been clicked in the past, and (ii) users with the same information need tend to submit different textual queries even when seeking images that are conceptually similar. Exact match is therefore bound to produce a relative small number of samples per concept; so, we next propose methods with less stringent criteria for matching queries to concepts.

For each image, we use the terms in the queries for which the image has been clicked in order to create a surrogate textual description for that image (similar to [15]). This can then be viewed as a document (in the traditional IR sense) that can be indexed and retrieved in response to a query. To

this end, we employ a *language modelling* (LM) approach to IR [7]. In this approach, a language model φ_D is inferred for each document D . Given query Q , the documents are ranked by estimating the *likelihood of the query* $P(Q|\varphi_D)$. Queries are represented as sequences of k binary random variables each corresponding to a term, and the query likelihood is:

$$P(\mathbf{q}|\varphi_D) = P(q_1, q_2, \dots, q_k|\varphi_D) = \prod_{i=1}^k P(q_i|\varphi_D) \quad (1)$$

assuming that each q_i is generated independently from the previous ones given the document model. The language model is thus reduced to modelling the distribution of each single term. The simplest estimation strategy for an individual term probability is the *maximum likelihood estimate* (*mle*). This corresponds to the relative frequency of a term t_i in document d , $P_{mle}(t_i|\varphi_d) = \frac{tf_{i,d}}{\sum_t tf_{t,d}}$, where $tf_{i,d}$, the term frequency of term t_i in d , is normalised by the document’s length (the sum of the term frequencies of all of its terms). This method for selecting positive samples for concept c is denoted as *LM* when we use the concept name N_c as the query, and as *LM_{key}* when we use the concept name N_c together with the concepts’ keywords K_c as the query.

Eq. 1 assigns zero query likelihood probabilities to documents missing even a single query term. This sparse estimation problem is addressed by *smoothing* techniques, that redistribute some of the probability of the terms occurring in a document to the absent ones. We use a mixture model of the document model with a background model (the collection model in this case), well-known in text retrieval as Jelinek-Mercer smoothing [7]:

$$P(\mathbf{q}|\varphi_D) = \prod_{i=1}^k (1 - \lambda)P_{mle}(q_i|\varphi_D) + \lambda P_{mle}(q_i|\varphi_C) \quad (2)$$

where λ is a smoothing parameter (typically set to 0.8), and $P_{mle}(t_i|\varphi_C) = \frac{df_i}{\sum_t df_t}$, with df_i the document frequency of the term t_i in the collection. In our case the collection consists of the images that appear in the clickthrough data, i.e., images that have been clicked before for some query. In this work, we apply a variation of the above that requires that at least one of the query terms appears in the document. The selection method based on this smoothed LM is denoted as *LMS* when the concept name N_c is used as the query, and as *LMS_{key}* when the concept name N_c together with the concepts’ keywords K_c are used as the query.

The aim of these four LM-based selection strategies is to increase the number of positive samples by progressively relaxing the strictness of the matching criteria. This can be further achieved by applying stemming in each of these methods, resulting in *LM_{stem}*, *LM_{key_stem}*, *LMS_{stem}*, and *LMS_{key_stem}*, respectively. We use the open source *PF/Tijah* (<http://dbappl.cs.utwente.nl/pftijah/>) retrieval system [8] as the implementation of the above retrieval approaches.

The final technique we apply exploits the clickgraph in order to deal with the data sparsity and the possible mismatch of users’ query terms to concept names and keywords. The basic premise of this approach is that images clicked for the same query are likely to be relevant to each other, in the sense that their visual content is likely to pertain to similar semantic concept(s). For each concept c , we construct an initial image set that contains the images selected using the *exact* method. If this method does not produce

any results, we add the images clicked for the most textually similar query to the concept name (using LM as our retrieval model). This initial image set is then expanded with the images accessible by a 2-step traversal of the graph as follows. First, each image i in this initial set is added to a final set. For each such i , we first find the queries for which this image was clicked and then add to the final set, the images (other than the ones already there) clicked for that query. This method is denoted as *clickgraph* and produces a set of images. To rank these images, one approach is to apply this method after assigning weights to the edges of the clickgraph based on the number of clicks. Alternative approaches that exploit the clickgraph are iterative methods, such as the random walk models employed in [5].

Even though methods such as the ones described above aim to deal more effectively with the sparsity of the clickthrough data, they are likely to introduce false positives in the sample selection, i.e., images that were clicked but were not relevant. Given that our proposed methods produce a ranking of the images, a strategy to reduce this potential noise would be to filter the selected images by considering only those ranking above a given threshold. In this work, however, we do not apply such noise reduction techniques; we consider all retrieved images as positive samples (for the the LM-based methods, all samples with $P(Q|\varphi_D) > 0$ are selected as positive).

3.3 Negative sample selection

Negative samples are selected randomly. The probability of selecting a non-negative (i.e., positive) example in the original dataset \mathbf{I} is equal to the concept’s prior probability in \mathbf{I} , i.e., $P(c|\mathbf{I})$. Assuming that after positive sample selection the prior of c in the remaining set decreases, $P(c|\mathbf{I} - \mathbf{I}_{c,m}) \leq P(c|\mathbf{I})$, then the prior $P(c|\mathbf{I})$ is an upper bound for the probability of error. Random negative sample selection will therefore be accurate for rare concepts.

The number of negative examples has to be sufficient for training (e.g., description of the class boundaries in minimum margin classifiers). At the same time, though, it should not be too high, since that would lead to an increase in the number of false negatives. In this work, we arbitrarily select the number of negatives to be $N_{c,m} = \max(1000 - N_{c,m}, N_{c,m})$, where $N_{c,m} = |\mathbf{I}_{c,m}|$ is the number of positive examples for c . With this approach, the training set for any concept contains at least 1000 samples in total. In case the number of positive examples is high (above 500), then the number of negative examples increases accordingly, so that enough samples are available for the possibly more complex classification/ranking problem that arises.

3.4 Automatic image annotation

For each image in the collection, two types of low-level features are extracted, one capturing visual information in the image and another based on text captions accompanying the images. Both features are similar to the ones used in [18]. Text features are required since some concepts cannot be described using visual features only (e.g., “war”). Using features based on text allows the evaluation of the generated training sets for these concepts. In any case, however, relevance judgments are based on visual appearance and not image metadata.

For the visual description, the Integrated Weibull distribution [20] is extracted from a number of overlapping image

regions. The region distributions are then compared against the distributions of images belonging to a set of common reference concepts (or proto-concepts). This leads to a $120 - d$ feature vector \mathbf{F}_W .

For the text-based feature vector, a vocabulary of the most frequently used words is built for each concept, using the available textual metadata. Each image caption is compared against each concept vocabulary and a frequency-histogram $\mathbf{F}_{T,c}$ is built for each concept c . The feature vector length is equal to the vocabulary size, but is usually very sparse due to the short length of typical captions.

For ranking with classifiers, each image is represented by its (visual or text) feature vector and the score output by a support vector machine (SVM) classifier. The classifiers employ an RBF kernel and 3-fold cross-validation is performed on the training set to select the class weight parameters $w+$ and $w-$. The LibSVM [3] implementation is used as the basis of the classification system.

4. EXPERIMENTAL DESIGN

4.1 Datasets

The image collection **I** we use consists of 97,628 photos provided by *Belga News Agency* in the context of the activities of the VITALAS¹ project. The photographic images cover a broad domain, and can be characterised either as “editorial”, i.e., pictures with concrete content related to a particular event, e.g., sports, politics, etc., or as “creative”, i.e., pictures with artistic and timeless content, such as nature, work, etc. Each photo is accompanied by high quality metadata (defined by the IPTC) that include textual captions written manually by Belga’s professional archivists.

Belga also provided us with their search logs for a period of 101 days from June to October 2007. From these, we extracted the clickthrough data and performed a “light” normalisation on the text of the submitted queries, so as to clean up the data and identify identical/similar queries that had been submitted with slight variations. This preprocessing step included conversion to lower case and removal of punctuation, quotes, the term “and”, and the names of the major photo agencies that provide their content to Belga (e.g., EPA). The normalisation was deliberately kept shallow so that further steps, such as stemming and stopword removal, can be applied at a later stage where required. These search log data contain 35,894 of the images that also belong to **I** and which have been clicked for 9,605 unique queries. Given that Belga is a commercial portal, their search log data are much smaller in size, compared to those collected, for instance, by a general purpose search engine [5]. On the other hand, given that it provides services to professional users, mainly journalists, we expect their search log data to be relatively less noisy. The sparsity of the clickthrough data is evident, though, similarly to [5], in the power-law distributions observed for the images-per-query and queries-per-image pairs (figures not included due to space limitations).

The VITALAS project has developed a multimedia concept lexicon which currently stands at around 500 entries. These concepts have been selected following a multi-step process involving a statistical analysis of Belga’s image captions [14], feedback by Belga’s professional archivists, and the addition of concepts from MediaMill [18] and LSCOM

[1]. Out of these, we selected 25 concepts for our experiments (see Table 1) based on various criteria, including the availability of search log-based positive samples and whether they are generalisable across collections. We also aimed to include a large number of sports-related concepts, given that 38.8% of the images in **I** have been classified as belonging to the IPTC “sport”. Given the manual annotations described next, we also aimed to include concepts with high variation in their frequencies in the manually annotated sets.

Table 1: The list of the 25 concepts used in our experiments together with their keywords

Concept c		
id	name	keywords
1	airplane_flying	air
2	airport	plane, runway
3	anderlecht	sport, soccer, football, club, belgian
4	athlete	sport
5	basketball	nba, competition, team, dribbling, passing, sport, player
6	building	
7	club_brugge	soccer, football, game, match, breydel, player, club bruges, dexia, belgian
8	crowd	mass, event, protest, demonstration, people
9	farms	agricultural, people, field, countryside
10	fashion_model	
11	fire	red flames, warm, fireman, firefighter
12	flood	rain, river
13	formula_one	f1, ecclestone, ferrari, mclaren, bmw, raikkonen, hamilton
14	highway	road, freeway, superhighway, autoroute, autobahn, expressway, motorway
15	logo	
16	meadow	sheep, goats, grass, field
17	rally_motorsport	motor, racing
18	red_devils	sport, soccer, football, belgian
19	sky	clouds, sun, moon
20	soccer	football
21	stadium	sport, game, match, competition, athleticism, stands, tracks
22	team	group
23	tennis	racket, court, match
24	volleyball	volley, ball, net, beach
25	war	

A large-scale manual annotation effort has been undertaken by Belga staff for the images in collection **I**. The presence of the VITALAS concepts was assumed to be binary. This process has yielded an incomplete, but reliable ground truth. For our selected 25 semantic concepts c , their manual annotation sets contain between 994 and 1000 annotated samples.

Given that this work is based on the assumption that the queries for which an image was clicked provide in essence an annotation of the image’s visual content, we perform a brief analysis to examine the extent to which this assumption holds, i.e., whether the positive samples selected by the search log based methods proposed in Section 3.2 can be considered as reliable annotations. To assess that, we compare, for each concept, the positive samples selected by each of our proposed methods against the manual annotations for that concept (a similar analysis has been performed by [11] for the case of clicks collected from a textual search engine).

Table 2 presents the results of this analysis. The numbers in the brackets correspond to the number of positive samples selected by our search log based methods that overlap with the set of manual annotations for that concept. The agreement percentage is defined as the ratio of the number of positive samples that agree with the manual annotations

¹<http://vitalas.ercim.org/>

Table 2: Agreement between positive samples selected by search log based methods and manual annotations

	$\mathbf{I}_{c,exact}$	$\mathbf{I}_{c,LM}$	$\mathbf{I}_{c,LMS}$	$\mathbf{I}_{c,LMS_{key}}$	$\mathbf{I}_{c,LM_{stem}}$	$\mathbf{I}_{c,LMS_{stem}}$	$\mathbf{I}_{c,LMS_{stem}_{key}}$	$\mathbf{I}_{c,clickgraph}$
airplane_flying		0.4286 (7)	0.4286 (7)	0.4444 (9)		0.4286 (7)	0.4444 (9)	0.2500 (4)
airport	0.0000 (1)	0.7755 (49)	0.7755 (49)	0.7600 (50)	0.7755 (49)	0.7755 (49)	0.7600 (50)	0.5556 (9)
anderlecht	0.6782 (289)	0.6796 (309)	0.6796 (309)	0.6327 (343)	0.6796 (309)	0.6796 (309)	0.6327 (343)	0.6140 (399)
athlete				0.0000 (1)	0.5000 (4)	0.5000 (4)	0.4000 (5)	
basketball	1.0000 (4)	0.8182 (11)	0.8182 (11)	0.9000 (20)	0.8182 (11)	0.8182 (11)	0.9000 (20)	1.0000 (4)
building		0.2727 (11)	0.2727 (11)	0.2727 (11)	0.2727 (11)	0.2727 (11)	0.2727 (11)	
club_brugge	0.9636 (55)	0.9459 (74)	0.9179 (134)	0.9068 (161)	0.9459 (74)	0.9179 (134)	0.9068 (161)	0.8016 (252)
crowd				0.0000 (2)			0.0435 (23)	
farms				0.9032 (31)		0.9032 (31)	0.9032 (31)	
fashion_model			0.9032 (31)	0.9032 (31)		0.9032 (31)	0.9032 (31)	
fire	0.3333 (3)	0.5000 (32)	0.5000 (32)	0.4848 (33)	0.5750 (40)	0.5750 (40)	0.5682 (44)	0.5000 (52)
flood	0.9487 (39)	0.8906 (64)	0.8906 (64)	0.8974 (78)	0.9000 (110)	0.9000 (110)	0.9076 (119)	0.9032 (93)
formula_one	1.0000 (6)	0.9333 (15)	0.8571 (21)	0.8072 (83)	0.8571 (21)	0.8571 (21)	0.8072 (83)	0.8298 (47)
highway	1.0000 (2)	1.0000 (10)	1.0000 (10)	1.0000 (12)	1.0000 (12)	1.0000 (12)	1.0000 (14)	1.0000 (2)
logo	1.0000 (3)	1.0000 (12)	1.0000 (12)	1.0000 (12)	1.0000 (12)	1.0000 (12)	1.0000 (12)	1.0000 (14)
meadow				0.0000 (3)			0.0000 (13)	
rally_motorsport		0.7857 (14)	0.7857 (14)	0.7857 (14)	0.7857 (14)	0.7857 (14)	0.7857 (14)	0.6000 (5)
red_devils	0.9787 (94)	0.9833 (120)	0.9690 (129)	0.9441 (161)	0.9833 (120)	0.9690 (129)	0.9441 (161)	0.8794 (257)
sky	0.0000 (1)	0.3333 (6)	0.3333 (6)	0.2857 (7)	0.3333 (6)	0.3333 (6)	0.2857 (7)	0.0000 (1)
soccer	0.2381 (21)	0.5714 (42)	0.5714 (42)	0.5682 (44)	0.5714 (42)	0.5714 (42)	0.5682 (44)	0.6707 (167)
stadium		0.8750 (8)	0.8750 (8)	0.8750 (8)	0.8750 (8)	0.8750 (8)	0.7778 (9)	1.0000 (4)
team		0.1667 (6)	0.1667 (6)	0.1667 (6)	0.1667 (6)	0.1667 (6)	0.1667 (6)	0.5000 (2)
tennis	1.0000 (1)	0.7500 (8)	0.7500 (8)	0.8333 (12)	0.7500 (8)	0.7500 (8)	0.8333 (12)	0.6923 (78)
volleyball	0.8929 (28)	0.7763 (76)	0.7763 (76)	0.7564 (78)	0.7662 (77)	0.7662 (77)	0.7468 (79)	0.6738 (141)
war		1.0000 (3)	1.0000 (3)	1.0000 (3)	1.0000 (3)	1.0000 (3)	1.0000 (3)	1.0000 (2)

to the overlap size. Missing value indicates zero overlap. As expected the number of samples increases with smoothing, stemming, and addition of keywords. In the case of the *clickgraph* method, there is no apparent trend; the number of samples appears to be very concept-specific. Overall, the number of samples that overlap with the manual annotations exhibits a high variability across concepts and employed methods, with a mean of 51.1 and a median of 14.

Table 2 indicates that the level of agreement between manual and search log based annotations varies greatly across concepts. On the one extreme, concepts “highway”, “logo”, and “war” have 100% agreement, albeit for a small number of samples, whereas “meadow” has a 0% agreement, a result which is to be expected though given that there is one manually annotated positive sample. About half the concepts have a level of agreement higher than 70%, i.e., 70% of the samples selected using search log based methods are true positives. On the other hand, concepts such as “fire”, “athlete”, “building”, “farms”, “sky”, “team”, “airplane_flying” have around or more than 50% false positives, indicating that their automatically selected training sets are likely to be noisy.

4.2 Description of experiments

We performed four types of experiments for evaluating the effectiveness of using search log based methods to select images to be used as alternative or complementary data sources to manual annotations for training concept classifiers. Our experimental setting refers to the search logs as ‘SL’, the manually annotated set as ‘MA’, and the common evaluation set (defined in experiment two) as ‘CE’.

Exp. 1: SL training, MA evaluation (feasibility test).

For this first experiment, we build the classifiers using the training data originating from the search logs $\mathbf{T}_{c,m}$ (as described in Section 3) that do not overlap with the manual annotations. Image representation is based on the \mathbf{F}_W feature only, so only visual information is used. For results to be comparable across the different positive sample selec-

tion methods m , the negative sample set is the same across all datasets of each concept. Effectiveness is measured on the data already manually annotated by Belga’s archivists. This allows us to directly compute the evaluation metrics, without performing any manual assessments, but results in each concept having its own evaluation set. This experiment serves the following purposes: (i) it is a first indication on the usefulness of the training sets $\mathbf{T}_{c,m}$, and (ii) from an application standpoint, it demonstrates how re-ranking of a conventional text query can be performed using classifier systems trained using automatically generated data.

Exp. 2: SL training, CE evaluation.

The common evaluation set for experiment 2 (and also experiments 3 and 4) is obtained after removing all manual and automatically generated sets from the original image set \mathbf{I} . Hence, the set $\mathbf{I}_{eval} = \mathbf{I} - \bigcup_{i,j} \mathbf{T}_{c_i,m_j}$ contains 56,605 images.

Note that for a given concept c , the randomly selected negative examples are common for all methods m (except for the manual method which does not have any randomly selected samples). In this experiment the training sets are generated using the search logs (i.e., no manual annotations are used) and evaluation is performed on the common evaluation set. Visual or text-based low level features are used.

Exp. 3: SL & MA training, CE evaluation.

This experiment uses the training sets of experiment 2, but combines them with the manually annotated data. Hence new training sets are generated for each non-manual method m , such that $\mathbf{T}'_{c,m} = \mathbf{T}_{c,m} \cup \mathbf{T}_{c,manual}$. If an image belongs to both $\mathbf{T}_{c,m}$ and $\mathbf{T}_{c,manual}$ the manual annotation takes priority. Note that both the randomly selected and the manually annotated negative examples are used. Evaluation is again performed on the common evaluation set \mathbf{I}_{eval} and classification uses either visual or text-based low-level features.

Exp. 4: MA training, CE evaluation.

This is a baseline experiment, where only manual annotations are used to train the classifiers, which are then evalu-

ated on \mathbf{I}_{eval} first for visual and then for text-based features. Generally, manual annotations are expected to provide the best results since they contain the most accurate and reliable assessments resulting in training sets of higher quality.

5. EXPERIMENTAL RESULTS

Results for the first experiment (feasibility test) are directly produced, since the existing manual annotations are used as ground truth for the evaluation. Table 3 shows the results in terms of the average precision (AP) attained for each concept and training set generation method (this is averaged over ten runs so as to avoid bias due to random negative sample selection). These results indicate that: (i) in most cases, the AP value is considerably higher than the prior, (ii) training set generation methods based on language modelling tend to perform better than the exact match approach, leading to the conclusion that the additional samples obtained are useful, and (iii) the clickgraph method performs worse than the LM approaches, despite the increased number of samples; this can be attributed to the noise that this method introduces.

For experiments 2, 3 and 4, ground truth is not available for the common evaluation set. In order to assess the ranking performance, the authors manually annotated for each concept the set created by pooling the top 200 results of all experiments for that concept. As evaluation metric, we use the precision at the first 200 results ($P@200$). Table 4 presents the maximum $P@200$ achieved for each concept with the corresponding training set generation method. This table and especially the results of experiment 2 provide a confirmation of our previous indication that the language modelling methods produce better results than the exact match and clickgraph approaches. In addition, experiment 3 generally improves the results over ones obtained from the manual annotations. The contribution from the search log-based training data is therefore positive.

Figure 1 shows a detailed example for one training set generation method. Figure 2 provides the mean $P@200$ across all concepts for all features and training set generation methods, and allows the following interesting observations. (i) For \mathbf{F}_W , the automatically generated training data alone (exp. 2) cannot surpass the performance of the manually produced ones (exp. 4). (ii) Combining the two training data sources, however, consistently gives the best results (exp. 3). (iii) Surprisingly, the use of $\mathbf{F}_{T,c}$ in experiment 2 results in the less noisy methods (the ones not involving keywords or the clickgraph) producing better results compared to methods based on the inclusion of manual annotations (exp. 3 and 4). (iv) Regarding the comparison between the low-level features, $\mathbf{F}_{T,c}$ dominates, but this is to be expected. Examination of the results per concept, however, reveals that in some cases (e.g., concept *sky*), \mathbf{F}_W achieves better performance. This is typically observed for concepts strongly associated with the image content, rather than the image context. Figure 3 provides a more qualitative view of the results by illustrating samples taken from the manual and automatically generated training sets, as well as the results for a run that uses \mathbf{F}_W .

Readers are invited to visit <http://olympus.ee.auth.gr/~diou/civr2009/> to view the image lists returned for each concept and method combination, along with the training set used and the performance achieved.

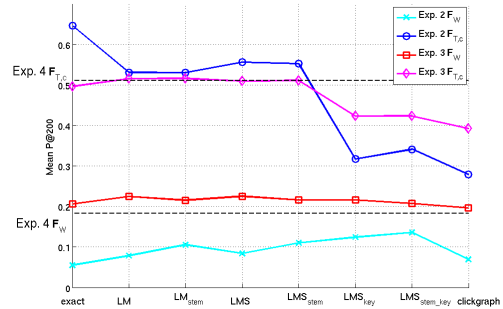


Figure 2: Mean of $P@200$ across all concepts for all experiments and training set generation methods.

6. CONCLUSIONS & FUTURE WORK

This paper demonstrated how clickthrough data can contribute to reducing the effort required to create and/or maintain the training data needed for automatic image annotation using supervised learning. We expect our results to enable the practical application of the ‘detector approach’ to annotation, which reduces the investment required to apply image annotation in ‘the real world’. Existing content owners can create concept detectors specialised to their domain by simply exploiting the usage logs - or start collecting these right away! The main advantages of our approach grounded in clickthrough data are its scalability in the number of concept detectors, and the possibility to dynamically adapt the detector set, automatically keeping track of concepts that change or emerge. Our experiments show that the idea is feasible in a commercial setting with professional content users. An open question is whether our positive results transfer to the more noisy settings of web image search for general (non-professional) use.

ACKNOWLEDGMENTS

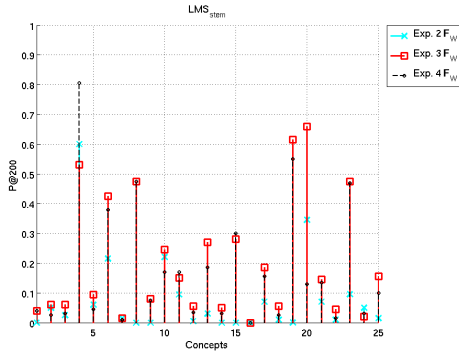
The authors are grateful to the Belga press agency for providing the images and search logs used in this work and to Marco Palomino from the University of Sunderland for the extraction of the text features used. This work was supported by the EU-funded VITALAS project (FP6-045389). Christos Diou is supported by the Greek State Scholarships Foundation (<http://www.iky.gr>).

7. REFERENCES

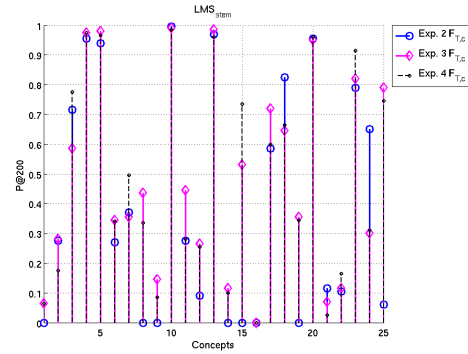
- [1] LSCOM Lexicon Definitions and Annotations Version 1.0. Technical report, Columbia University, 2006.
- [2] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza. Improving search engines by query clustering. *Journal of the American Society for Information Science and Technology*, 58(12):1793–1804, 2007.
- [3] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [4] S.-F. Chang, J. He, Y.-G. Jiang, E. El Khoury, C.-W. Ngo, A. Yanagawa, and E. Zavesky. Columbia University/VIREO-CityU/IRIT TRECVID2008 high-level feature extraction and interactive video search. In *Proc. of TRECVID 2008*, 2008.

Table 3: Feasibility test, average precision results.

Concept c	$T_{c,exact}$	$T_{c,LM}$	$T_{c,LMS}$	$T_{c,LMS_{key}}$	$T_{c,LM_{stem}}$	$T_{c,LMS_{stem}}$	$T_{c,LMS_{stem}_{key}}$	$T_{c,clickgraph}$	prior
airplane_flying				0.0589			0.0589		0.0262
airport	0.2269	0.3736	0.3736	0.3637	0.3736	0.3736	0.3637	0.3032	0.2181
anderlecht	0.5920	0.6003	0.6003	0.5496	0.6003	0.6003	0.5501	0.5168	0.3223
athlete				0.3419	0.4547	0.4547	0.4859		0.3968
basketball	0.5172	0.5499	0.5499	0.5473	0.5499	0.5499	0.5521	0.5172	0.3855
building		0.2166	0.2166	0.2166	0.2166	0.2166	0.2166	0.0779	0.1034
club_brugge	0.5353	0.5786	0.5224	0.6082	0.5786	0.5224	0.6056	0.5030	0.4080
crowd				0.3854			0.3854		0.1494
farms				0.0623			0.0677		0.0090
fashion_model			0.7227	0.7227		0.7116	0.7116		0.4333
fire	0.3868	0.4523	0.4523	0.3371	0.4395	0.4395	0.3424	0.3620	0.0972
flood	0.5159	0.5794	0.5794	0.5333	0.5948	0.5948	0.4511	0.4261	0.3627
formula_one	0.4322	0.5478	0.5890	0.7242	0.5813	0.5813	0.7242	0.7257	0.4208
highway				0.3062			0.3223		0.1623
logo								0.5453	0.4322
meadow				0.0162			0.0162		0.0010
rally_motorsport		0.6899	0.6899	0.7197	0.6899	0.6899	0.7018	0.5556	0.2763
red_devils	0.7540	0.7837	0.8046	0.7191	0.7977	0.8046	0.7191	0.6584	0.4624
sky				0.2662			0.2448		0.1454
soccer	0.5694	0.6435	0.6435	0.6475	0.6435	0.6435	0.6475	0.5600	0.4297
stadium		0.3954	0.3954	0.2169	0.3954	0.3954	0.1913		0.1091
team		0.2153	0.2153	0.1905	0.2153	0.2153	0.2153	0.0675	0.0371
tennis	0.4471	0.5044	0.5044	0.5016	0.5044	0.5044	0.5016	0.4588	0.3717
volleyball	0.5561	0.5678	0.5678	0.5119	0.5678	0.5678	0.5119	0.4541	0.3403
war		0.1737	0.1737	0.1737	0.1737	0.1737	0.1737	0.3120	0.2076
MAP	0.5030	0.4920	0.5059	0.4050	0.4928	0.5022	0.4067	0.4402	0.2523



(a)



(b)

Figure 1: An example of the $P@200$ values attained for all concepts with $T_{LMS_{stem}}$. Concept numbers correspond to Table 1. $P@200 = 0$ for both features indicates that the corresponding concept has not been evaluated due to insufficient training data for method LMS_{stem} .

[5] N. Craswell and M. Szummer. Random walks on the click graph. In *Proc. of the 30th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 239–246, 2007.

[6] A. Hauptmann, R. Yan, and W.-H. Lin. How many high-level concepts will fill the semantic gap in news video retrieval? In *Proc. of the 6th International Conference on Content-based Image and Video Retrieval (CIVR 2007)*, pages 627–634, 2007.

[7] D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In *Proc. of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL 1998)*, pages 569–584, 1998.

[8] D. Hiemstra, H. Rode, R. van Os, and J. Flokstra. PF/Tijah: text search in an XML database system. In *Proc. of the 2nd International Workshop on Open Source Information Retrieval (OSIR 2006)*, pages 12–17, 2006.

[9] J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, and X. Zhang, editors. *Proc. of the 17th International Conference on World Wide Web*, 2008.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of the 8th Annual International ACM SIGKDD Conference on Knowledge Discovery and Data mining*, pages 133–142, 2002.

[11] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Transactions on Information Systems*, 25(2), 2007.

[12] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[13] A. Natsev, W. Jiang, M. Merler, J. R. Smith, J. Tešić,



Figure 3: Positive examples from the manual annotations, an automatically generated training set using the LMS_{stem_key} method, and the first 12 results for F_W in experiment 2. All images ©Belga (please refer to <http://olympus.ee.auth.gr/~diou/civr2009/belga.html> for the full copyright notice).

Table 4: Maximum $P@200$ value achieved and the corresponding training set generation method. The results that reached or exceeded the baseline results of the manual annotations are highlighted.

Concept c	Experiment 2		Experiment 3		Experiment 4	
	F_W	$F_{T,c}$	F_W	$F_{T,c}$	F_W	$F_{T,c}$
airplane_flying	0.0250 (LMS _{key})	0.0650 (LMS _{key})	0.0400 (exact)	0.0700 (LMS _{key})	0.0400	0.0650
airport	0.0500 (LM)	0.2850 (LMS _{key})	0.0650 (LMS _{key})	0.2800 (LM)	0.0250	0.1750
anderlecht	0.0350 (LMS _{key})	0.7150 (LM)	0.0600 (exact)	0.6000 (exact)	0.0300	0.7750
athlete	0.6000 (LM _{stem})	0.9550 (LM _{stem})	0.7650 (LMS _{key})	0.9750 (exact)	0.8050	0.9750
basketball	0.0600 (LM)	0.9400 (LM)	0.1050 (LMS _{stem_key})	0.9850 (exact)	0.0450	0.9650
building	0.2150 (LM)	0.2700 (LM)	0.4250 (LM)	0.3450 (exact)	0.3800	0.3400
club_brugge	0.0300 (LM)	0.4400 (LM)	0.0150 (LM)	0.4900 (LM)	0.0100	0.4950
crowd	0.4950 (LMS _{key})	0.3200 (LMS _{key})	0.4750 (exact)	0.4550 (LMS _{key})	0.4750	0.3350
farms	0.1250 (LMS _{stem_key})	0.1350 (LMS _{stem_key})	0.0800 (exact)	0.1450 (exact)	0.0750	0.0850
fashion_model	0.2200 (LMS _{stem})	1.0000 (LMS)	0.2500 (exact)	0.9900 (LMS)	0.1700	0.9850
fire	0.1150 (exact)	0.3100 (exact)	0.1700 (exact)	0.4450 (LM)	0.1700	0.2800
flood	0.0150 (LMS _{key})	0.1500 (LMS _{key})	0.0650 (LMS _{key})	0.2650 (exact)	0.0350	0.2550
formula_one	0.1600 (LMS _{key})	0.9700 (LM)	0.2800 (exact)	0.9850 (LM _{stem})	0.1850	0.9600
highway	0.0200 (LMS _{key})	0.1100 (LMS _{stem_key})	0.0500 (exact)	0.1150 (exact)	0.0300	0.1000
logo	0.3750 (clickgraph)	0.2900 (clickgraph)	0.2800 (exact)	0.5800 (clickgraph)	0.3000	0.7350
meadow	0.0950 (LMS _{key})	0.0250 (LMS _{key})	0.0800 (LMS _{key})	0.0200 (LMS _{key})	0.0000	0.0000
rally_motorsport	0.0800 (clickgraph)	0.6750 (LMS _{key})	0.1950 (LMS _{stem_key})	0.7200 (LM)	0.1550	0.6000
red_devils	0.0300 (LM _{stem})	0.8250 (LM _{stem})	0.0550 (LMS)	0.6450 (LMS)	0.0250	0.6650
sky	0.3950 (LMS _{stem_key})	0.1050 (LMS _{key})	0.6150 (exact)	0.3550 (exact)	0.5500	0.3450
soccer	0.5450 (LMS _{key})	0.9800 (LMS _{key})	0.6650 (LMS _{key})	0.9650 (exact)	0.1300	0.9600
stadium	0.0700 (LM)	0.1150 (LM)	0.1550 (exact)	0.0800 (exact)	0.1350	0.0250
team	0.0150 (LMS _{key})	0.1250 (LMS _{key})	0.0450 (exact)	0.1400 (exact)	0.0150	0.1650
tennis	0.1150 (clickgraph)	0.8450 (LMS _{key})	0.4900 (LMS _{key})	0.8300 (exact)	0.4700	0.9150
volleyball	0.0700 (exact)	0.6500 (LM)	0.0300 (LMS _{key})	0.3150 (LMS _{key})	0.0300	0.3100
war	0.0150 (LM)	0.0600 (LM)	0.1550 (LM)	0.8200 (exact)	0.1000	0.7450

L. Xie, and R. Yan. IBM Research TRECVID-2008 Video Retrieval System. In *Proc. of TRECVID 2008*, 2008.

[14] M. A. Palomino, M. P. Oakes, and T. Wuytack. Automatic extraction of keywords for a multimedia search engine using the chi-square test. In *Proc. of the 9th Dutch-Belgian Information Retrieval Workshop (DIR 2009)*, pages 3–10, 2009.

[15] B. Poblete and R. A. Baeza-Yates. Query-sets: using implicit feedback and query patterns to organize Web documents. In Huai et al. [9], pages 41–50.

[16] A. S. and G. Quénot. Video corpus annotation using active learning. In *Proc. of the 30th European Conference on IR Research*, pages 187–198, 2008.

[17] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In Huai et al. [9], pages 327–336.

[18] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proc. of the 14th ACM International Conference on Multimedia*, pages 421–430, 2004.

[19] A. Ulges, M. Koch, C. Schulze, and T. Breuel. Learning TRECVID’08 high-level features from YouTubeTM. In *Proc. of TRECVID 2008*, 2008.

[20] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, C. G. M. Snoek, and A. W. M. Smeulders. Robust scene categorization by learning image statistics in context. In *International Workshop on Semantic Learning Applications in Multimedia*, page 105, 2006.

[21] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. of the SIGCHI Conference on Human factors in computing systems*, pages 319–326, 2004.

[22] X.-J. Wang, W.-Y. M. Ma, and X. Li. Exploring statistical correlations for image retrieval. *Multimedia Systems*, 11(4):340–351, 2006.

[23] J. Yang and A. G. Hauptmann. (un)reliability of video concept detection. In *Proc. of the 7th International Conference on Content-based Image and Video Retrieval (CIVR 2008)*, pages 85–94, 2008.