What Can Pictures Tell Us About Web Pages? Improving Document Search using Images

Sergio Rodriguez-Vaamonde[†] [*]
sergio.rodriguez@tecnalia.com

Lorenzo Torresani[§]
lorenzo@cs.dartmouth.edu

Andrew Fitzgibbon[¶]
awf@microsoft.com

[†]Tecnalia, Zamudio, Spain & University of the Basque Country, Bilbao, Spain
[§]Dartmouth College, Hanover, NH, U.S.A.
[¶]Microsoft Research Cambridge, United Kingdom

ABSTRACT

Traditional Web search engines do not use the images in the HTML pages to find relevant documents for a given query. Instead, they typically operate by computing a measure of agreement between the keywords provided by the user and only the text portion of each page. In this paper we study whether the *content* of the pictures appearing in a Web page can be used to enrich the semantic description of an HTML document and consequently boost the performance of a keyword-based search engine. We present a Web-scalable system that exploits a pure text-based search engine to find an initial set of candidate documents for a given query. Then, the candidate set is reranked using semantic information extracted from the images contained in the pages. The resulting system retains the computational efficiency of traditional text-based search engines with only a small additional storage cost needed to encode the visual information. We test our approach on the TREC 2009 Million Query Track, where we show that our use of visual content yields improvement in accuracies for two distinct text-based search engines, including the system with the best reported performance on this benchmark.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

Keywords: Web Search, Ranking, Image Content

1. INTRODUCTION

"A picture is worth a thousand words." Despite this old saying, modern Web search engines ignore the pictures in HTML pages and retrieve documents merely by comparing the query keywords with the text in the documents. Of course this text includes the words in image captions and markup tags, but does not look at the pixels themselves. The exclusive reliance on text-based technology to search the Web is explained by the challenges posed by the handling of image data: automatic image understanding is still today computationally expensive and prone to mistakes.

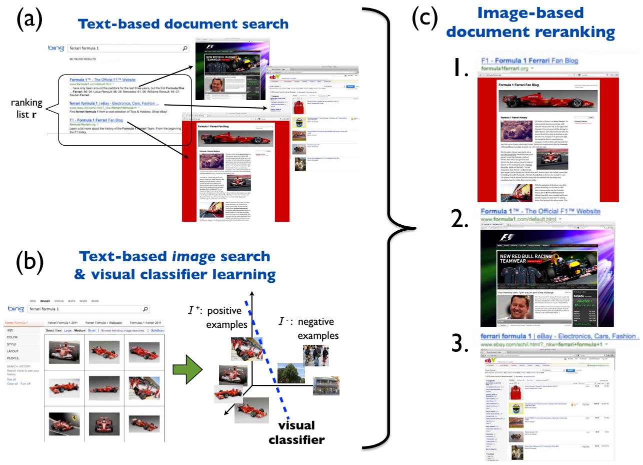
(a)
Text-based document search
ranking list r
(b)
Text-based *image* search & visual classifier learning
I^+: positive examples
I^-: negative examples
visual classifier
(c)
Image-based document reranking
1.
2.
3.

Figure 1: Method overview: the query q is issued (a) to a document search engine producing a ranked list r of Web pages and (b) to a text-based image search engine yielding positive image examples to learn a query-specific visual classifier. Finally, (c) the visual classifier is used to rerank the pages in the list r.

In this paper we propose a novel document retrieval approach that uses the content of the pictures in the Web pages to boost the accuracy of pure text-based search engines. At a high-level we expect that, for example, for the query "Ferrari Formula 1", users will judge documents containing pictures of Ferrari cars to be more relevant than pages with unrelated images. Consequently our hope is that a search system combining the textual information with the visual information extracted from the pictures will yield improved accuracy. While there is a large literature on combining text and image data for *image* search, we know of no work that attempts to improve *document* search using image content. The closest work to ours is the approach of Yu et al. [7] who demonstrate improved ranking by using simple image measures such as aspect ratio, size, and high-level features such as blurriness. In contrast, we use a modern object recognition system to provide rich data on the image content.

[*] This research was conducted while the first author was a visiting student at Dartmouth College.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.
Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...$15.00.

2. APPROACH OVERVIEW

In order to design an image-based search engine that can scale to Web-size databases we are posed with two fundamental challenges. First, the descriptor extracted from the pictures must be semantically rich but also very compact so that the overall size of the document is sufficiently small for fast search in billions of pages. Second, we must devise a way to efficiently translate the query keywords into a visual model (i.e., an image classifier) that can be used to measure the compatibility between the text query and the photos in a Web page. We address the first requirement by utilizing a compact attribute-based image descriptor—the classeme vector [6]—which has been shown to yield accurate object recognition even with simple linear classifiers, which are efficient to train and test. The second requirement is met by learning "on the fly" the visual model associated to the query keywords using as positive training examples the top image results of a text-based image search engine, such as Google Images or Bing Images. The visual classifier can then be used together with the text-based techniques of traditional Web search to measure the compatibility between the query and the page content, now both *visual* as well as *textual*.

The architecture of our system is illustrated in Fig. 1. Let $\mathcal{D}$ be the database of Web pages. In order to produce the list of relevant documents for an input query q, we use a reranking strategy combining traditional text-retrieval methods with the visual classifier learned for query q:

(a) The query q is provided as input to a text-based search engine $\mathcal{S}$ operating on $\mathcal{D}$ to produce a ranking list $\mathbf{r}$ of K candidate pages (Fig. 1(a)).

(b) In parallel, the query q is issued to a keyword-based *image* search engine (in this work we use the visual search service of Bing Images). The top M image results $\mathcal{I}^+$ are used as positive examples to train a visual classifier to recognize the query concept in images (Fig. 1(b)). As negative training set $\mathcal{I}^-$, we use a fixed collection of images representative of many object classes.

(c) The list of pages $\mathbf{r}$ is reranked (Fig. 1(c)) by taking into account several image features including the classification scores produced by evaluating the visual classifier on the pictures of the K candidate pages.

The intuition is that when the query represents a concept that can be recognized in images, the learned visual classifier can be applied to increase or decrease the relevancy of a candidate page in the ranking list depending on whether the document contains pictures exhibiting that visual concept.

Our system can perform efficient query-time learning and testing of the visual classifier in large databases. This scalability stems from the small size of the classeme vector (only 333 bytes/image) and the use of a linear (i.e., fast to train and test) classification model. Here we use a linear Support Vector Machine (SVM) trained on $M = 50$ examples.

3. AN IMAGE-BASED MODEL FOR DOCUMENT RERANKING

We now describe our image-based reranking model. We use a query-relative representation of the documents: let $\mathbf{x}^{(q,i)} \in \mathbb{R}^d$ be the feature vector describing the i-th document in the database $\mathcal{D}$ relative to query q. Given an input query q, our approach enables real-time computation of the vector $\mathbf{x}^{(q,i)}$ for each document i in the ranking list $\mathbf{r}$ produced by text-search engine $\mathcal{S}$. The vector $\mathbf{x}^{(q,i)}$ includes several image-based features. In the next subsection we present our features. In subsection 3.2 we describe how these features are used to rerank the documents in $\mathbf{r}$.

3.1 The query-document features

The vector $\mathbf{x}^{(q,i)}$ for query-document pair (q, i) comprises the following 12 features.

Text features $(\mathbf{x}_{1,2}^{(q,i)})$: *'relevance score'* and *'ranking position'* of document i in the ranking list $\mathbf{r}$ produced by $\mathcal{S}$ for query q. The *'relevance score'* feature is a numerical value indicating the relevancy of the document as estimated by $\mathcal{S}$, purely based on text. The *'ranking position'* is the position of i in the ranking list $\mathbf{r}$. By including these two features we leverage the high-accuracy of modern text-based search.

Visual metadata features $(\mathbf{x}_{3,4}^{(q,i)})$: *'# linked images'* and *'# valid images'*. These attributes are used to describe whether the document contains many images. Web pages often include many small images corresponding to clipart, icons and graphical separators. These images usually do not convey semantic information. To remove such images from consideration, we extract the classeme vector only from pictures having at least 100 pixels per side. The feature *'# valid images'* gives the total number of images in the page for which the classeme descriptor was computed.

Query visualness features $(\mathbf{x}_{5,6}^{(q,i)})$: *'visual classifier accuracy'* and *'visual concept frequency'*. These features are dependent only on the query (i.e., they are constant for all documents) and describe the ability of the visual classifier learned for query q to recognize that concept in images. In particular, *'visual classifier accuracy'* is the 5-fold cross-validation accuracy of the classifier trained on the examples retrieved by Bing Images for query q. While this feature describes how reliably the classifier recognizes query q in images, it does not convey how frequently this visual concept is present in pictures of Web pages. This information is captured by *'visual concept frequency'* which is the fraction of times the visual classifier for query q returns a positive score on images of the database $\mathcal{D}$.

Intuitively, these two query visualness features provide the reranker with an indication of the usefulness of employing the visual classifier for query q to find relevant pages.

Visual content features $(\mathbf{x}_{7-12}^{(q,i)})$: *'histogram of visual scores'* and *'document relevancy probability'*.

The *'histogram of visual scores'* is a 5-bin histogram $(\mathbf{x}_{7-11}^{(q,i)})$ representing the quantized distribution of the scores (i.e., the SVM outputs) produced by the visual classifier of query q on the images of document i.

The *'document relevancy probability'* $(\mathbf{x}_{12}^{(q,i)})$ is the posterior probability that the document i is relevant for query q given the observed classification scores of the images contained in the page, i.e., $p(i \text{ is relevant } | s_1, \ldots, s_{n_i})$, where $s_1, \ldots, s_{n_i}$ are the binarized scores that the SVM for query q produces on the n_i (valid) images of document i. This probability is computed via standard Bayes's rule under the assumption of conditional independence (the Naïve Bayes assumption):

$$p(i \text{ is relevant } | s_1, \ldots, s_{n_i}) = \\ p(i \text{ is relevant }) TP^{m_i} (1 - TP)^{n_i - m_i} / p(s_1, \ldots, s_{n_i}) \quad (1)$$

where m_i is the number of images of i having positive classification score while TP denotes the true positive rate of the classifier, i.e., $TP = p(s_u = 1 | i \text{ is relevant })$. The denominator in Eq. 1 can be evaluated via application of the sum and product rules in terms of the prior, TP, and the false positive rate (FP). We assume that the rates TP, FP are query-independent and we estimate them empirically over a large number of labeled training queries.

3.2 Learning to rerank using visual content

Our objective is to learn a reranking function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $f(\mathbf{x}^{(q,i)})$ provides a numerical estimate of the final relevancy of document i for query q, where i is one of the pages in the list $\mathbf{r}$ retrieved by $\mathcal{S}$. In order to avoid the computational cost of training the reranker at query-time, we learn a *query-independent* function f: this function is trained only once during an *offline* training stage, using a large collection of labeled training examples for many different queries. We denote with $\mathcal{T} = \{(q_1, \mathbf{r}_1, \mathbf{y}_1), \dots, (q_N, \mathbf{r}_N, \mathbf{y}_N)\}$ the offline training set used to learn f, where $\mathbf{r}_j$ is the sorted ranking list of K documents produced by the text-based search engine $\mathcal{S}$ for input query q_j, i.e., $r_{jk} \in \mathcal{D}$ denotes the ID of the document ranked in the k-th position; the vector $\mathbf{y}_j$ contains the corresponding ground-truth relevance labels. We use binary relevance labels with $y_{jk} = 1$ denoting that document r_{jk} is relevant for query q_j, and value 0 indicating "non-relevant". We denote with $\boldsymbol{\theta}$ the learning parameters of the function, i.e., $f(\mathbf{x}^{(q,i)}) = f(\mathbf{x}^{(q,i)}; \boldsymbol{\theta})$. In our experiments we tested the following reranking models:

- **Ranking SVM.** This algorithm [4] learns a linear model of the features, i.e., $f(\mathbf{x}^{(q,i)}; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x}^{(q,i)}$. The parameters $\boldsymbol{\theta}$ are optimized to produce a ranking that preserves as much as possible the ordering of the training examples, i.e., such that ideally $\boldsymbol{\theta}^T \mathbf{x}^{(q_j,k)} > \boldsymbol{\theta}^T \mathbf{x}^{(q_j,l)} \iff y_{jk} > y_{jl}$.
- **Random Forest.** This method learns a random forest [2] with each tree greedily optimized to predict the relevance labels y_{jk} of the training examples. The resulting hypothesis computes an average of the P independently trained regression trees $f^{(1)}, \dots, f^{(P)}$, i.e., $f(\mathbf{x}^{(q,i)}; \boldsymbol{\theta}) = \frac{1}{P} f^{(p)}(\mathbf{x}^{(q,i)})$. The P trees are diversified by considering at each split only $d' < d$ randomly chosen features (we set d' to 10% of the number of features). The value of P is selected via cross-validation.
- **Gradient Boosted Regression Trees (GBRT).** This model also predicts by averaging the outputs of P regression trees. However, unlike in case of the random forest where the trees are independently learned, the GBRT trees are trained in sequence to correct the current regression error (for further details see [9]).

4. DISCUSSION OF COSTS

Although our implementation requires downloading the images returned by the image search engine and then extracting the classeme vectors from them, in a real application scenario the classeme descriptors (which are query-independent) would be precomputed at the time of the creation of the index by the image-search service. Then the image and document search would be issued in parallel, and the image service would return only the classeme vectors for the image results (333 bytes per image). The computational cost of learning the query-specific visual classifier on the classeme vectors is certainly of the same order as ranking in existing text-based systems. Finally, testing the visual classifier is also efficient: it takes less than one second to evaluate a linear SVM on 1M classeme vectors.

		p@10	p@30
$\mathcal{S}$=UDMQ	Ranking w/ text only ($\mathcal{S}$)	48.2	38.8
	Our method w/ Ranking SVM	48.3	38.7
	Our method w/ Random Forest	53.2	32.5
	Our method w/ GBRT	64.5	40.5
$\mathcal{S}$=Indri	Ranking w/ text only ($\mathcal{S}$)	27.7	27.7
	Our method w/ Ranking SVM	27.8	27.3
	Our method w/ Random Forest	31.6	23.4
	Our method w/ GBRT	37.3	27.2

Table 1: Precision @ 10 and 30 on the TREC MQ09 benchmark using different ranking models. Top: search engines based on UDMQ. Bottom: search engines based on Indri. Our GBRT reranker using image features achieves consistently the best accuracy and greatly outperforms the engines using text only (UDMQ and Indri).

As for the storage cost, our system requires saving the classeme vectors of the valid images in each Web page. In the dataset used for our experiments, each page contains on average 1.44 valid images. Thus, the added storage cost due to the use of images is less than 500 bytes per document, which can be easily absorbed by modern retrieval systems.

5. EXPERIMENTS

We evaluate our system on the ad-hoc retrieval benchmark of the TREC 2009 Million Query Track (MQ09) [3]. This benchmark is based on the "Category B" ClueWeb09 dataset [1] which includes roughly 50 million English pages crawled from the Web. The publicly available distribution of this dataset includes the original HTML pages collected by the ClueWeb09 team in 2009, but not the images linked in them. In order to run our image-based system on this collection, in September 2011 we attempted to download the pictures linked in these documents. Unfortunately many of the pages and images were no longer available on the Web. Thus here we restrict our experiments only to the pages for which we successfully downloaded *all* images linked in the original document (this amounts to 41% of the pages).

To train and test our reranking system, we use the publicly available MQ09 queries and human relevance judgements.In all, judgements are available for 684 queries, with each query receiving either 32 or 64 document assessments. The relevance values are "not relevant" ($y_{jk} = 0$) or "relevant" ($y_{jk} = 1$). In order to meet the conditions for reusability of the MQ09 topics and judgements [3], we chose as our text-search engines $\mathcal{S}$ the UDMQAxQEWeb system [8], which was one of the systems participating in the MQ09 competition. We refer to this system as UDMQ. The ranking lists of UDMQ on the MQ09 queries are publicly available.

To test the ability of our method to work with different text-search systems $\mathcal{S}$, we also present results with the popular Indri search engine [5]. We generated the ranking lists of Indri on the MQ09 queries by using its public batch query service. Unlike UDMQ, Indri did not participate to the MQ09 competition. Thus, while the estimate of the *absolute* accuracy of Indri on MQ09 may be unreliable, here we use it just as a baseline to judge the *relative* improvement produced by reranking its search results with our system.

For both engines, we generate the vector $\mathbf{r}$ by truncating the ranking list at $K = 200$. We employ 10-fold cross validation over the queries, thus using in each run 9/10th of the queries for training and the remaining 1/10-th for validation. Performance is measured as precision at 10 and 30 (denoted as statMPC@10 and statMPC@30) using the

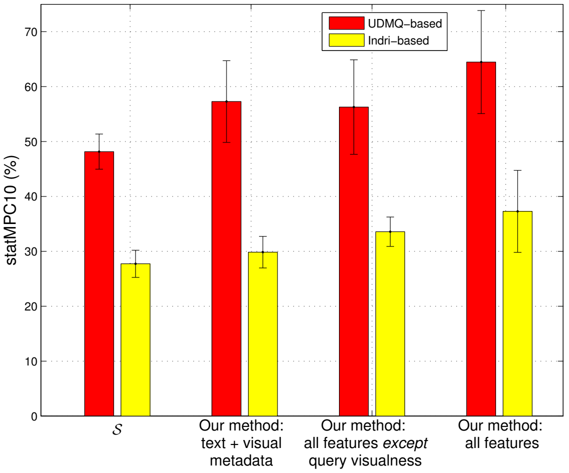

Figure 2: Precision @ 10 using different image features with the GBRT reranker based on UDMQ (red) and Indri (yellow). Removing the visual content features ("text + visual metadata") or the query visualness features from our descriptor causes a large drop in performance.

"statistical evaluation method" [3] . We focus on these measures as our main goal is to improve the relevancy of the documents in the top part of the ranking list.

In Table 1 we compare the accuracy of the text-based search engines (UDMQ and Indri) to the different image-based ranking models introduced in section 3.2. First, we see that all image-based rerankers yield higher values of statMPC@10 than the search engines using text only. The GBRT reranker is by far the best, improving by over 33% the precision of UDMQ, which achieved the highest accuracy among all search engines participating in the MQ09 competition. This clearly indicates that our image-based features provide new and relevant information compared to that captured by traditional text-based engines. Instead, no significant gain is achieved in terms of statMPC@30. Empirically we found that our reranker tends to apply fairly small displacements to the positions of documents in the original ranking list. While these small rearrangements have a positive impact on the top-10 lists examined by statMPC@10, they are too small to change sensibly the statMPC@30.

Next, we want to study which features contribute to the statMPC@10 improvement. For this purpose we retrain the GBRT model (our best performing model) using two different variants of our feature vector: 1) "text + visual metadata" (i.e., we use only the features $\mathbf{x}_{1-4}^{(q,i)}$, which do not capture the content of the images); 2) the vector "all features *except* visualness" (i.e., we *exclude* only features $\mathbf{x}_{5,6}^{(q,i)}$, which capture the document-independent visualness of the query). The results are presented in Figure 2 using UDMQ (red bars) and Indri (yellow bars) as text-retrieval models $\mathcal{S}$. We see that, although GBRT with the "text + visual metadata" descriptor achieves accuracy slightly superior to the text-based search engines, the performance is not as good as when our approach uses all features, *including* the visual content. This suggests that despite the noisy nature of the Bing training images, our visual classifier does capture information that is useful to predict whether a document is relevant with respect to the query. Excluding the query visualness features from our descriptor also causes a drop in accuracy. Intuitively, this happens as these features allow the reranker to determine whether the query is visually recognizable and to modulate accordingly the contribution of the visual content features in the reranking function.

		% of queries	median gain in p@10	median visual error
$\mathcal{S}$=UDMQ	$\mathcal{S}$ wins	15.3	20.0	29.4
	GBRT wins	12.6	33.1	25.7
	tie	72.1	n/a	27.6
$\mathcal{S}$=Indri	$\mathcal{S}$ wins	12.6	20.0	27.7
	GBRT wins	14.5	29.5	25.2
	tie	72.9	n/a	28.4

Table 2: A comparison across queries between the text-based engines and our GBRT image-based reranker. Note that the "median visual error" (i.e., the cross-validation error of the visual classifier) is higher for the queries where $\mathcal{S}$ wins compared to the queries where our approach wins: this suggests that our method does better when the query is more visual.

In Table 2 we report the percentage of queries for which our image-based GBRT reranker provides a higher value of prec@10 than $\mathcal{S}$, i.e., "wins" over the text-based engine. Our method and $\mathcal{S}$ are tied for roughly 72% of the queries, while the number of times one wins over the other are fairly evenly divided. However, in the cases where our system wins, it gives a much higher gain in prec@10, compared to when $\mathcal{S}$ wins (+33.1% vs +20% when $\mathcal{S}$=UDMQ; +29.5% vs +20% when $\mathcal{S}$=Indri). It is also interesting to observe that the cross-validation error of the visual classifier is lower for the subset of queries where our system wins over $\mathcal{S}$.

6. CONCLUSIONS

In this work we have studied the largely unexplored topic of how to improve Web-document search using images. We have demonstrated that by using modern object recognition systems it is possible to extract useful semantic content from the photos of a Web page and that this additional information improves the accuracy of state-of-the-art text-based retrieval systems. All this is achieved at the small cost of a few additional hundred bytes of storage for each page.

7. ACKNOWLEDGMENTS

This work was supported in part by Microsoft Research and by NSF CAREER award IIS-0952943. SRV's visit to Dartmouth College was partially funded by the Basque Government under grant number IE11-316.

8. REFERENCES

[1] Website. `http://lemurproject.org/clueweb09.php/`.

[2] L. Breiman. Random forests. *Machine Learn.*, 45(1):5–32, 2001.

[3] B. Carterette, V. Pavlu, H. Fang, and E. Kanoulas. TREC Million Query Track 2009 Overview. 2009.

[4] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, 2000.

[5] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: a language-model based search engine for complex queries. In *Proc. of ICIA*, 2005.

[6] L. Torresani, M. Szummer, and A. W. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010.

[7] Q. Yu, S. Shi, Z. Li, J.-R. Wen, and W.-Y. Ma. Improve ranking by using image information. In *Proc. of ECIR*, 2007.

[8] W. Zheng and H. Fang. Axiomatic Approaches to Information Retrieval - TREC 2009 Million Query and Web Tracks. 2009.

[9] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A General Boosting Method and its Application to Learning Ranking Functions for Web Search. In *NIPS*, 2007.