Snap-and-Ask: Answering Multimodal Question by Naming Visual Instance *

Wei Zhang, Lei Pang, Chong-Wah Ngo
Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
{wzhang34, leipang3}@student.cityu.edu.hk, cscwngo@cityu.edu.hk

ABSTRACT

In real-life, it is easier to provide a visual cue when asking a question about a possibly unfamiliar topic, for example, asking the question, "Where was this crop circle found?". Providing an image of the instance is far more convenient than texting a verbose description of the visual properties, especially when the name of the query instance is not known. Nevertheless, having to identify the visual instance before processing the question and eventually returning the answer makes multimodal question-answering technically challenging. This paper addresses the problem of visual-to-text naming through the paradigm of answering-by-search in a two-stage computational framework, which is composed out of instance search (IS) and similar question ranking (QR). In IS, names of the instances are inferred from similar visual examples searched through a million-scale image dataset. For recalling instances of non-planar and non-rigid shapes, spatial configurations that emphasize topology consistency while allowing for local variations in matches have been incorporated. In QR, the candidate names of the instance are statistically identified from search results and directly utilized to retrieve similar questions from community-contributed QA (cQA) archives. By parsing questions into syntactic trees, a fuzzy matching between the inquirer's question and cQA questions is performed to locate answers and recommend related questions to the inquirer. The proposed framework is evaluated on a wide range of visual instances (e.g., fashion, art, food, pet, logo, and landmark) over various QA categories (e.g., factoid, definition, how-to, and opinion).

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous

*Area chair: Frank Nack

MM'12, October 29–November 2, 2012, Nara, Japan.

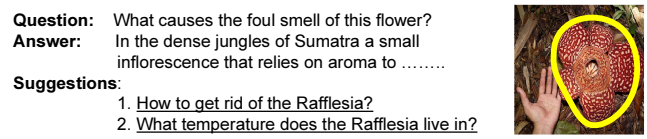

Figure 1: A text-based question referring to a picture. The target "Rafflesia" is first named through instance search. The answer is then retrieved from the cQA website, together with some suggested questions.

General Terms

Algorithms, Performance, Experimentation

Keywords

Multimedia question answering, visual instance search, similar question search

1. INTRODUCTION

While intensive research efforts have been devoted to multimedia search [12], multimedia question-answering (MQA) in broad domains is still a relatively new and largely untapped research area. With the convergence of mobile and social media computing, MQA is predicted to play a more prominent role that is complementary to general search for two main reasons. First, mobile devices are emerging as the major instrument used to access the Internet. Due to small form factors and limited battery power/bandwidth, browsing a long list of search results, especially multimedia items, is not as convenient as on desktop machines. Instead, the majority of mobile users expect a short list of answers that directly address their concern. With rich sensors embedded in mobile devices, it has also become natural for mobile users to ask a question through speech, or by texting a short sentence with an image example as a reference. Second, the proliferation of social media suggests that MQA could be tackled by leveraging the massive amount of multimedia data accumulated online for answering questions. Furthermore, for certain categories of questions (e.g., how-to), it is more intuitive to answer the questions with multimedia content than with pure text description [14].

This paper addresses a practical scenario in MQA: a user snaps a picture and asks a question referring to the picture. Figure 1 shows an example of a text-based question referring to a picture with the instance "Rafflesia". Multimodal questions are different from text-based QAs where, rather than

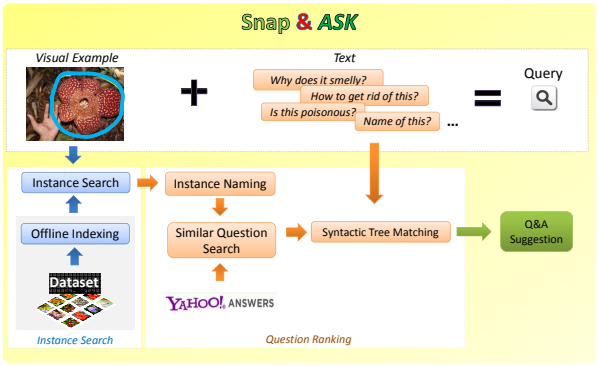

Figure 2: Framework for Snap-and-Ask.

providing a long textual description for the visual instance in a question, visual examples, which are more direct and intuitive, are given, possibly with the target instance masked by the inquirer. The returned result is a short ranked list of QA pairs crawled from community-contributed QA (cQA) archives that best match the inquirer's request. The ranked list not only serves as the right answer, but also recommends some related questions, similar in spirit to query suggestion [10] in commercial search engines.

Figure 2 shows an overview of the proposed framework for MQA, which is composed out of two major components: instance search (IS) and question ranking (QR). Given a multimodal question, the provided image example is first treated as visual query and searched against a million-scale Web image dataset. This search scenario is different from typical image or near-duplicate search in the following two aspects: the retrieved target items could appear in different background contexts – a task that is different from near-duplicate search [25, 27]; and the target could be an object of non-planar surface or non-rigid shape (e.g., flag, food, animal) – a task that commercial mobile search engines [3], such as Google Goggles[1], cannot deal with. Our proposed technique considers the use of spatial topology for enhancing the robustness and scalability of the matching. The candidate names of an instance are then extracted from the metadata of the retrieved items through syntactic and semantic analysis of natural language. Each candidate name is then treated as a query to the cQA website "Yahoo! Answers"[2] for retrieving similar questions. Due to the fact that questions given by various users could be phrased in wildly different ways, the pool of questions obtained from multiple queries are subsequently parsed into syntactic trees, and matched against the inquirer's question. To this end, a small subset of QA pairs is selected, ranked, and finally presented to the user.

The main contribution of this paper is the proposal of a novel framework for addressing a practical scenario of MQA, where a mobile user snaps a picture and texts a short question while on the move. In contrast to other existing works, which assume text-based questions [1, 16, 26], and operate in a specific domain (e.g., news [28], product [29], cooking [13]), the framework, grounded on the answering-by-search paradigm, deals with questions in broad domains, covering different types of instances and question categories. Additionally, the framework also tackles problems in large-scale search of visual instances, and the matching of community-based questions by automatically naming the visual instance of the inquirer's question. The former remains an open problem in the literature, while the latter has not been previously attempted in the domain of multimedia QA. The remaining of the paper is organized as follows. Section 2 describes related works in MQA, as well as the state-of-the-art techniques in instance, near-duplicate, and mobile search. Section 3 presents the proposed technique for visual instance search, by considering the spatial configuration. Section 4 describes our algorithm for naming visual instances, while Section 5 details the parsing, matching, and ranking of similar questions. Section 6 presents experimental results, and finally Section 7 concludes the paper.

[1]http://www.google.com/mobile/goggles/
[2]http://answers.yahoo.com/

2. RELATED WORK

The proposed work is rooted in various content based retrieval techniques, ranging from query-by-example [12], multimodal query processing (e.g., search with text plus visual examples in TRECVID [22]), and data-driven search [24]. However, these techniques mostly deal with non-question-oriented queries, and are thus different from our work. In the following, we mainly summarize the related works in two areas: multimedia answering and visual instance search.

Most existing works in *multimedia answering* are extended from text-based QA systems, which aim to find multimedia content (e.g., images and videos) as answers to text questions in a specific domain. In the news domain, VideoQA [28] is an early system, which supports factoid QAs by leveraging visual content, ASR (Automatic Speech Recognition) transcripts, and online information for locating news video segments as answers. In the documentary domain, a passage retrieval algorithm, which is based on video caption recognition and ranking, was proposed in [26] to return passages associated with short video clips as answers. In the domain of educational video, QA was investigated by [1] based on lexical pattern matching on ASR transcripts and PowerPoint slides. Basically these systems rely heavily on rich and relatively clean textual information extracted from ASR transcripts and video captions. Applying these techniques directly to UGC (user generated content) from the Internet is difficult, due to the poor accuracy of speech and caption recognition. Recently, in the domain of consumer electronics, a community-based QA system supporting how-to QAs was developed for retrieving Web videos as answers [14]. The system adopts a two-step approach: first it develops concept classifiers for visual recognition, and second it detects concepts from user questions by searching for similar questions on a cQA website. The two sets of concepts are then compared for ranking video answers. While interesting, this work is not easily extended to other domains for the requirement to train a large number of domain-specific visual classifiers, which remains practically difficult based on current technology. More recently, the problem of media selection for generating answers was also addressed by [16], where given a text-based QA pair, the media type (text, image, or video) that will best answer the question is predicted. Different from the proposed work, all the aforementioned papers consider text-based questions only. The most similar work compared to our work is Photo-based QA [29], which also considers multimodal questions (text and photo). The system first performs visual matching of the query photo

Figure 3: Challenging examples for instance search.

with images from a database, and then answers are extracted from a QA repository named START[3] by text-based template matching. Although similar in spirit, Photo-based QA only considers factoid QAs, and the use of template matching also limits the types of questions that can be answered. Furthermore, issues of instance search, such as scalability when using a large image dataset, and matching robustness against non-planar and non-rigid objects, are not addressed.

Visual instance search (IS) aims to find multimedia examples that contain the same target, but not necessarily with the same background context as the query. A recent pilot study in TRECVID [22] indicated that the search task could be challenging when the target instance exhibits variations in size, viewpoint, and resolution under different backgrounds. The utilization of spatial layout and context for efficient indexing and searching remains unexplorered [11]. A close relative of IS is near-duplicate (ND) search [25, 27], where the matching is performed at the full image level. ND has received numerous research attention recently, and the existing techniques can be grouped into several categories. First, feature point matching with bag-of-visual-words, supported by inverted file indexing, is performed, followed by strong geometric verification for the top few retrieved items [19]. Second, auxiliary information, such as the position, scale, orientation, and signature of words, is also indexed for fast filtering of false matches by weak geometric verification [8]. Third, spatial information, such as spatially close feature points, are indexed for constraint matching [27, 32]. Some of these techniques have been successfully adopted for mobile media search in different vertical domains, such as CD, book cover, and location search [4], but not on instances with non-planar surfaces and non-rigid shapes.

3. VISUAL INSTANCE SEARCH

There are two peculiarities, when an inquirer provides an image as reference for a text question. First, the query picture will be snapped with care, and hence with acceptable visual quality. It is reasonable to assume that an inquirer may retake the same instance multiple times, such that pictures with artifacts (e.g., motion blur and occlusion) will not be issued. Second, with the support of multi-touch technologies on mobile devices, inquirers can easily draw a mask on a query image to distinguish the instance-of-interest from background context.

While queries could be assumed to be generated in a relatively clean setting, the challenge of instance search, however, originates from the wide range of instances that can possibly be taken as queries. With reference to Figure 3, the difficulties can be briefly summarized as: *visual variation*, *spatial constraint*, and *context utilization*. As shown in Figure 3(a), the instance candidates to be retrieved from a reference dataset could appear in wildly different forms, especially for 3D objects, under different lighting conditions and capturing viewpoints. Furthermore, certain instance types, such as fashion and art, can be rich of repeated patterns as in Figure 3(b), in addition to being non-rigid 3D instances. Matching instances with a linear transformation model, as adopted in near-duplicate search [19], is no longer suitable. Finally, instances snapped outdoors, for example, the "Bruce Lee Statue" in Figure 3(c), might benefit from using the background context for searching. However, this assumption can not be generalized to other instances such as fashion and logos. This section addresses the former two difficulties, while leaving the last problem as our future work.

3.1 Retrieval Model

Our model is grounded on the recent advances in bag-of-visual-words representation (BoW) [21]. Initially a large vocabulary tree of one million leaf visual words is constructed from the public image dataset MIRFLICKR [6]. The implementation is based on [17], where local features (SIFT) are clustered by k-means hierarchically in a top-down manner, and a branching factor of 100 is used to split each non-leaf node. During the offline indexing, SIFT descriptors extracted from Web images are parsed down the tree until they reach the leaf nodes that best match the descriptors. Through this step, descriptors are quantized to the nearest visual word, and indexed into an inverted file structure for fast online retrieval. Auxiliary information, including the signatures and the spatial locations, are also indexed for word filtering and geometric verification. The signatures, represented as a binary vector of 32 bits, are generated by Hamming embedding (HE) [8]. During online retrieval, a similar procedure is carried out for processing the instance-of-interest marked by the inquirer. To alleviate the adverse effect due to the quantization error, a descriptor is assigned to multiple visual words by soft-weighting [20]. By traversing the index file with HE filtering, images sharing common visual words with the query are rapidly retrieved from the reference dataset. Finally, spatial topology consistency is taken into account for better image ranking.

3.2 Spatial Configuration

As background context is not utilized, the information content of a query is thereby reduced. Modeling the spatial configuration of visual instance for geometric verification becomes crucial for an effective search. The existing models, which rely on a set of linear transformations, impose either too weak or too strong constraints for instance matching. For example, WGC (weak geometric consistency) [8], which efficiently votes for the dominant differences of scale and orientation among matched visual words, only weakly enforces spatial consistency, and does not guarantee the local regularity (e.g., relative position) of correspondences. On the other hand, spatial verification [19], which warps the matched visual words between images to discover the dominant linear transformation, imposes strict constraint, and only works well for near-duplicate scenes, and planar or rigid objects.

[3]http://start.csail.mit.edu/

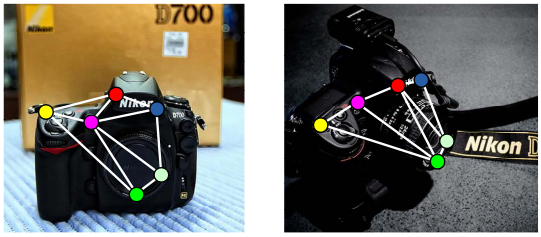

Figure 4: Construction of triangulation meshes based on the matching visual words between two images. The matched words are indicated with the same color.

3.2.1 Matching by sketching meshes

We model the spatial information by seeking a compromised model, based on Delaunay Triangulation (DT) [2, 9], which is neither too weak to identify the spatial inconsistency nor too strong to rule out the positive spatial configurations. DT is a technique used in computer graphics for building meshes out of a point set, such that no point is inside the circumcircle of any triangles. For instance search, given the matched words between a query instance $\mathcal{Q}$ and a reference image $\mathcal{R}$, DT sketches the spatial structures of $\mathcal{Q}$ and $\mathcal{R}$ respectively based on the matches. Figure 4 shows an example of triangulation meshes constructed from matched visual words of $\mathcal{Q}$ and $\mathcal{R}$. With $\Delta\mathcal{Q}$ denoting the mesh of $\mathcal{Q}$, the geometric consistency of $\mathcal{R}$ and $\mathcal{Q}$, named *bonus factor*, is measured as:

$$\mathbf{BF}(\mathcal{Q},\mathcal{R}) = \log(N+1) \times \exp\{\mathrm{Sim}_{DT}(\Delta\mathcal{Q},\Delta\mathcal{R})\}, \quad (1)$$

where N is the total number of matched features by traversing the inverted file. Sim_{DT} measures the percentage of common edges[4] between $\mathcal{Q}$ and $\mathcal{R}$:

$$\mathrm{Sim}_{DT}(\Delta\mathcal{Q},\Delta\mathcal{R}) = \frac{\mathbf{E}_{\Delta\mathcal{Q}} \cap \mathbf{E}_{\Delta\mathcal{R}}}{\mathbf{E}_{\Delta\mathcal{Q}}}, \quad (2)$$

where $\mathbf{E}_{\Delta\mathcal{Q}}$ denotes the edge set of $\Delta\mathcal{Q}$. In Equation (1), note that Sim_{DT} is weighted by a factor of $\log(N+1)$, so that an image, sharing more matched visual words with $\mathcal{Q}$, receives a higher **BF** value. For constructing meshes, the one-to-one mapping constraint needs to be enforced. This is done by allowing a point from $\mathcal{Q}$ to match another point on $\mathcal{R}$ with the smallest Hamming distance. The enforcement effectively prevents an excessive number of redundant matches, a problem known as the "burstiness" effect [7], which could corrupt similarity scores when there are repeated patterns in images.

While simple, DT has the following merits: (1) the relative spatial position of words is considered, (2) no assumption of any transformation model is made, (3) a certain degree of freedom for variations of word positions is allowed. Compared to WGC [8], criterion (1) considers the topology of words, and thereby is more effective in measuring geometric consistency. Compared with strict spatial verification [31], criterion (2) does not impose any prior knowledge on types of instances and transformations, and thus the checking of geometric coherency is looser. However, by allowing variations of local changes as stated by criterion (3) without the assumption of a transformation model, DT is a flexible model, which is more adaptable to non-rigid and non-planar instances under different capturing conditions. A fundamental difference between DT and other spatial verification methods is that no pruning of false matches or model estimation is involved. Instead, DT enumerates the potential true matches with the local topology consistency based on criteria (1) and (3), while tolerating false matches by not imposing any prior constraints based on criterion (2). Since DT acts positively in finding true matches rathar than negatively penalizing false matches, we name our measurement in Equation (1) the "bonus factor".

[4]Two edges are regarded as common if their vertices share the same visual words.

3.2.2 Computational Complexity

Two major steps of DT are the construction of triangulation meshes and the searching of common edges for Equation (2). The first step can be efficiently conducted by the divide-and-conquer method with a complexity of $O(n \log n)$, where n is the number of matched words between $\mathcal{Q}$ and $\mathcal{R}$. The second step can be done by a simple linear scan of edges with $O(|e|)$, where $|e| = O(n)$ is the number of edges. Basically, the computation is dominated by $O(n \log n)$. Based on our retrieval model, due to the use of the large vocabulary tree with one million words, n is usually a small number. In the case where the value of n is large, in our implementation, random sampling is performed, such that only a small subset of matches is evaluated by Equation (1). As DT is a "bonus model" which enumerates potential true matches, performance will not degrade severely with the down-sampling process.

4. NAME THE INSTANCE

Intuitively, given the search result, the name of an instance could be mined from the metadata based on statistical analysis, such as the term frequency (TF). Nevertheless, TF is measured at the word level, while a name is more appropriate to be represented at the phrase level. Furthermore, user-provided description could be inconsistent. For example, the recipe "beef broccoli" may be phrased as "broccoli beef" or "beef with broccoli". Directly applying TF could result in misleading statistics. This section presents the mining of instance names by phrase frequency, with the consideration of the potential noise in metadata. In addition, semantic similarity is also taken into account for reranking candidate names by random walk.

4.1 Noun Phrase Extraction

In general, an instance name could be a single word, an idiom, a restricted collocation, or a free combination of words. We regard a name as a noun phrase, and apply the Berkeley Parser[5] for producing the syntactic tree for each sentence in the metadata. The parser basically decomposes a sentence into units like subject, verb, and noun phrase, and then represents the relationship among units as a syntactic tree based on grammar formalism. In our case, we directly extract the subtrees, which correspond to the NP (noun phrase) units, from the tree as name candidates. However, a special case is when the metadata contains only one word in the title or the description. To boost recall, we also include these words as candidates. Note that we do not consider user tags here because some Web images (e.g., from Google image search) do not contain any tags.

[5]http://code.google.com/p/berkeleyparser/

4.2 Phrase Frequency

As a phrase could be more than one word, measuring the phrase frequency is not as straightforward as the word frequency. Given a phrase $\mathcal{P}_1$ composed out of one or multiple words, we measure the degree in which a second phrase $\mathcal{P}_2$ could be matched to $\mathcal{P}_1$ and thus contributes a score (≤ 1) to the phrase frequency (PF) of $\mathcal{P}_1$. This could be measured by counting the proportion of common words between $\mathcal{P}_1$ and $\mathcal{P}_2$. For example, if $\mathcal{P}_1$ is "Kony 2012" while $\mathcal{P}_2$ is "Kony", then $\mathcal{P}_2$ will contribute 0.5 to PF of $\mathcal{P}_1$. However, the degree of contribution should consider the source (i.e., the metadata and the image) $\mathcal{P}_2$ is extracted from. We consider two clues to model the noise level of the metadata. Generally speaking, the title of the metadata provides more reliable information than the description. Specifically, if the instance is the main subject of an image, the chance that the name of the instance appears in the title is also higher. Whereas for a description, the extracted name phrases are relatively diverse and likely to be contextually ralated rather than by content to the main subject (e.g., the name of a photographer, or the date of a publication). Second, the inherent noise level is also heuristically proportional to the document length, where the probability of a phrase corresponding to a correct name is higher if extracted from a brief description. For image-level noise, the score by instance search basically hints at the quality of a phrase extracted from an image. To this end, with $\mathbf{N}$ denoting the set of phrases extracted from the images retrieved by instance search, the frequency of a phrase $\mathcal{P}_i \in \mathbf{N}$ is defined as:

$$\mathrm{PF}(\mathcal{P}_i) = \sum_{\mathcal{P}_j \in \mathbf{N}, j \neq i} \frac{\alpha \times Sim^I(\mathcal{K}_j) \times Sim^S(\mathcal{P}_i, \mathcal{P}_j)}{\log D(\mathcal{P}_j)}, \quad (3)$$

where Sim^I returns the relevant score measured by instance search for the image, $\mathcal{K}_j$, where $\mathcal{P}_j$ is extracted from. The parameter α is empirically set to 1.0 if $\mathcal{P}_j$ is from the title of metadata and 0.3 otherwise. The function $D(\cdot)$ calculates the length of a document, used to penalize the phrases extracted from a lengthy description. The function Sim^S, measuring the syntactic relationship between two phrases, is defined as:

$$Sim^S(\mathcal{P}_i, \mathcal{P}_j) = \frac{|\mathcal{P}_i \cap \mathcal{P}_j|}{|\mathcal{P}_i \cup \mathcal{P}_j|}, \quad (4)$$

where $|\mathcal{P}_i|$ is the number of words in $\mathcal{P}_i$. Equation (4) basically measures the proportion of common words between two phrases by discounting ordering information. For example, the phrases "Kony 2012" and "2012 Kony" will be treated as the same phrase.

4.3 Re-ranking by Semantics

While Equation (3) suggests ranking noun phrases as name candidates by means of frequency, the relationship among phrases is not exploited. The set of noun phrases can be modeled as a graph, such that phrases can interact through information propagation by algorithms such as random walk [18]. The end result could lead to more realistic ranking: the significance of phrases can get boosted if actively communicating with peers, and otherwise diminished due to the isolation with other phrases. The major obstacle for constructing such a graph is the measurement of phrase relatedness, where existing similarity measures, such as the WordNet ontology, cannot be directly applied.

We propose an approach based on the work in [15] for measuring phrase similarity. Given two phrases, a vocabulary $\mathcal{V}$ is constructed, consisting of the number of distinctive words from them. A matrix $\mathcal{M}$ of size $|\mathcal{V}| \times |\mathcal{V}|$ is then formed, with each entry of $\mathcal{M}$ corresponding to the semantic similarity between two words in $\mathcal{V}$. By employing the technique described in [5], the similarity is measured based on the hypernym (is-a), meronymy (part-of), antonymy, and entailment relationships of the WordNet ontology. With $\mathcal{M}$, a phrase $\mathcal{P}$ is represented as a lexical semantic vector $\mathcal{L}$ of size $|\mathcal{V}|$, with each element $l_i = \max_{w_j \in \mathcal{P}} \mathcal{M}(i, j)$. The similarity between two phrases $\mathcal{P}_i$ and $\mathcal{P}_j$ is calculated as the cosine similarity between their vectors $\mathcal{L}_i$ and $\mathcal{L}_j$.

With the semantic similarity, we represent the phrase relationship as a graph, with phrases as nodes attributed by their frequencies, and pairwise semantic similarities between phrases as edges. By adopting random walk on the graph, the information is iteratively propagated among phrases till convergence, which results in a new ranked list of candidate names to be utilized for question matching.

5. QUESTION MATCHING AND RANKING

The cQA websites provide a good source of questions and answers manually crafted by human subjects. This section presents the techniques for finding similar questions (and hence answers) from online QA repositories. The general idea is by separately issuing the higher ranked candidate instance names as query to QA repositories, and then retrieving different sets of similar questions. This forms a pool of questions for further filtering and ranking by natural language processing, and eventually an inquirer is presented with the top ranked questions, which ideally contain the best answer and some suggested questions.

Ranking similar questions, however, is not trivial. For example, the questions "How to cook this dish?" and "Any tips for preparing this dish?" are similar, but share very few common keywords. We adopt the syntactic tree matching technique [23] to measure the question similarity. The technique divides the parse tree of a question into tree fragments, where each fragment captures the syntactic information of the question at a different granularity level based on grammar rules. The matching between two questions is carried out by recursively comparing the structures of tree fragments, while assigning higher weights to the matched fragments with larger sizes and deeper tree levels. In addition to the syntactic based matching, the words (leaf nodes of parse trees) of questions are also fuzzily matched based on WordNet's semantic similarity measure. This basically upgrades the similarity score for questions with different words but similar intention. We adopt this technique because no training is required, and furthermore, the similarity metric is tolerant to some grammatical errors, which are common on QA websites. In our implementation, when matching questions crawled from QA repositories, not only the question title but also the "content" of the question, which is provided by a user to detail the question, are compared. This will greatly improve the recall. For example, the question "What type of message is Kony 2012 trying to send?" could be matched to an inquirer's question "What does this poster mean?" if the content part has a description about the "Kony 2012" poster.

For multimodal QA, there are two strategies for matching an inquirer's question, where the instance name is phrased

Table 1: Dataset summary for multimodal questions.

QA categories	*Instance types*									*Total*
	fashion	food	pet	flower	art	product	logo	landmark	vehicle	
Factoid	19	8	9	12	13	9	7	10	9	96
Definition	10	5	5	4	6	1	5	8	3	47
How-to	14	20	9	8	6	23	4	1	5	90
Opinion	17	6	7	6	17	8	4	8	2	75
Total	60	39	30	30	42	41	20	27	19	308

as "this", with similar questions. The first strategy is to augment each parse tree with the candidate name before question matching by simply replacing a pronoun (e.g., "this") with the name. However, this strategy could be risky in producing misleading ranking of questions, when the candidate is not the actual name. For robustness, we only consider the top-1 ranked candidate name for this strategy. In other words, only the most confident candidate is employed for retrieving similar questions and augmenting the parse tree. The second strategy is a direct comparison between the original question and the retrieved similar questions without augmenting a candidate name. In this case, the pronoun "this" in a question will be treated as a noun phrase by the parser, and matched syntactically with similar questions. For this strategy, the top-k candidate names could be employed, so as to boost the recall of finding similar questions.

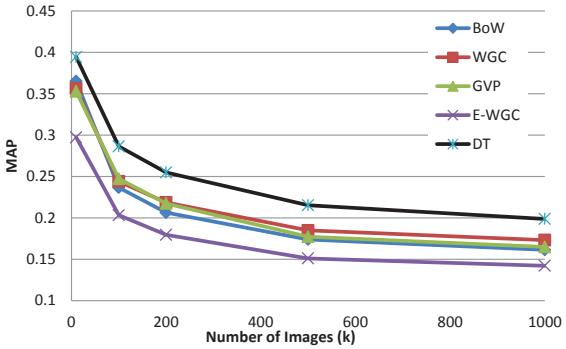

Figure 5: Instance search: Performance comparison of various approaches by varying the dataset size.

6. EXPERIMENT

6.1 Dataset Construction and Evaluation

Constructing a realistic MQA dataset is challenging, due to the lack of a publicly available dataset that consists of questions asked in the multimodal setting. Furthermore, the existing cQA archives are all text-based, which makes the dataset construction even harder. In Yahoo! Answers, for instance, users ask questions in text, and then have an option to provide further details (or question contents) by textually elaborating the questions or attaching hyperlinks to visual examples. To construct a test set for MQA, we first issued the names of instances as query keywords to Yahoo! Answers for retrieving the candidate questions. The returned results were usually mixed with irrelevant and redundant questions. We browsed through the results, and manually selected the sample questions that refer to the query instances. On the other hand, the same set of queries was also issued to Flickr and Google image search for crawling images as visual queries. Finally, the text-based questions were manually "transformed" into multimodal questions, by replacing the names of visual instances with the word "this", which refers to the image examples crawled from search engines.

To guarantee that a wide range of visual instances is considered in the test set, we issued a total number of 52 queries involving various instance types (e.g., fashion, art, pet, and etc.) to Yahoo! Answers, Flickr, and Google image search. Eventually, a test set of 308 questions was constructed, and there were six questions collected for each instance on average. Additionally, there were 438 images crawled as visual instances. For each of the queries, we randomly picked one image as the visual example for a question and as the query image for visual instance search. The remaining examples were then treated as the ground truth for the queries. The constructed questions were further grouped into four categories: factoid, definition, how-to, and opinion QAs. Table 1 summarizes the types of instances and questions in our dataset. Table 2, on the following page, lists a few sample questions in our dataset. To evaluate the scalability of the proposed approach for instance search, we also constructed a dataset including one million recent-uploaded pictures crawled from Flickr as distracting images, in addition to the 386 images as the ground-truth of the 52 instance queries.

For performance evaluation, we used the following metrics: MAP (mean average precision) for instance search; $MRR@K$ (mean reciprocal rank at rank K) for instance naming and question answering; and $P@K$ (precision at rank K) for question ranking. MRR measures the reciprocal rank of the first correctly returned item. The measure provides insights into the ability to return a correct instance name (or QA pair) at the top of the rankings. The metric P measures the proportion of related questions in a ranked list. Similar to MAP, MRR and P are averaged over all queries. The values of all metrics are in the range of [0,1], with 1 being the perfect, and 0 the worst performance.

6.2 Instance Search

We compared our approach, named DT, with the baseline BoW, GVP (geometric preserving visual phrases) [30], WGC (weak geometric consistency) [8], and E-WGC (enhanced WGC) [31]. For fair comparison, all the tested approaches were implemented upon the retrieval model described in Section 3.1, including the use of the large vocabulary tree, Hamming embedding, and multiple assignments. The major difference among them is the use of the spatial information. BoW does not impose any spatial constraints, while GVP is a voting approach that uses offset (or translation) information for rapid geometric checking. WGC, in contrast, utilizes the dominant scale and orientation voting for fast but weak

Table 2: Sample questions (*Ask*) referring to visual examples (*Snap*).

Instance type	*Ask*	*Snap and Name*
Fashion	1.What to wear with this boots?	
Food	2.How to make this?	
Pet	3.What's the breed of this dog?	
Flower	4.Why does this flower smell badly?	
Art	5.What does this mean?	
Product	6.How can I connect this to computer without ethernet?	
Logo	7.What is the meaning of this logo?	
Landmark	8.Who is this guy?	
Vehicle	9.Has anybody painted their car like this one? How?	

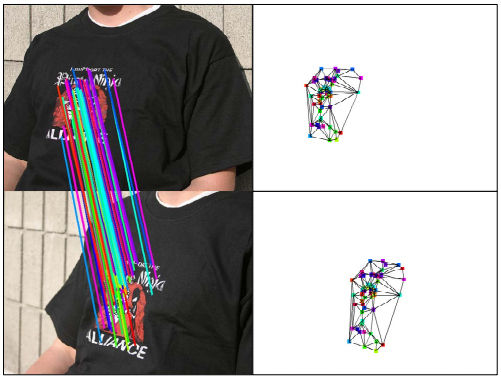

Figure 6: Meshes for non-planar instances: locally folded "Pirate Ninja Alliance" t-shirts.

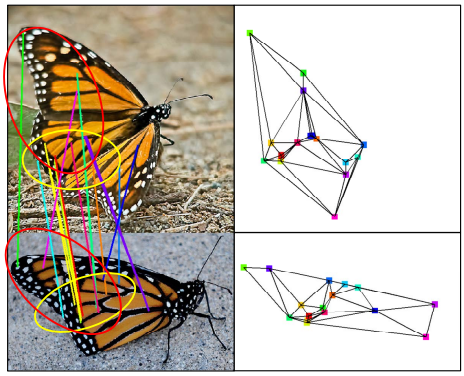

Figure 7: Meshes for non-rigid instances: "Monarch buttery" with flapping wings.

geometric verification. E-WGC incorporates the advantages of GVP and WGC by voting the translation after scale and orientation compensation.

Figure 5 shows the performance comparison in terms of *MAP* for the 52 queries, when the size of the dataset gradually increases from 0.01 million to 1.0 million Web images. Overall, DT consistently outperforms all other testing methods across different scales, and more importantly, the margin of improvement gets larger as the scale approaches one million. This result indicates the robustness and scalability of DT. We attribute the encouraging result to the merit of DT in effective topology consistency measurement, which tolerates local variations in matching, resulting in better ranking of candidate instances, especially for non-planar and non-rigid instances.

Referring to Figure 5, the performance is somewhat related to the degree of the transformation constraint imposed by each approach. Enforcing weaker constraints, such as GVP (2 degrees-of-freedom in spatial locations), exhibits slightly higher *MAP* than WGC (2 degrees-of-freedom in rotation and scale). E-WGC, which allows 4 degrees-of-freedom in depicting translation after scale and orientation compensation, results in the worst performance. Basically these approaches are sensitive to local perturbation of word matching on non-planar surfaces and non-rigid objects. E-WGC, which imposes stronger spatial constraints, tends to prune more true matches for these cases. With the use of the large vocabulary and Hamming Embedding, which already provide fine quantization for BoW matching, these approaches either exhibit a slight improvement (GVP and WGC) or a worse performance (E-WGC) than BoW. DT, in contrast, is not sensitive to local variations on word matching, as long as the perturbation does not alter the spatial topology or relative positions of the matched words. For instance, it is possible for DT to match the full regions of a locally folded t-shirt (Figure 6), but not the other approaches, which heavily prune potential true matches at the folded part. As a consequence, DT offers a higher capability (for non-rigid and non-planar instances) and tolerance (with the increase in the dataset size) in ranking, thus significantly improves the BoW baseline.

Figure 7 shows an example of matching the "Monarch butterfly" on two images. Owning to the non-rigid motion, the assumption of linear transformation is violated. E-WGC is only able to locate five out of ten true matches for similarity ranking. DT, which enforces the consistency of the spatial topology rather than a strict transformation, is able to accumulate evidence of matches from both wings (regions with yellow and red circles), and obtains a high similarity score of 0.67. In our dataset, there are plenty of man-made instances containing repeated patterns, such as the LV handbag and Arduino board. The matches by BoW are perturbed by these repeated patterns, and it is basically not wise to impose a strong spatial constraint to rule out these large numbers of perturbed but useful match patterns. DT measures topology consistency by enumerating the relative positions of words and usually ends up with a higher similarity than other approaches. In brief, DT benefits from being tolerant to local variations in BoW matches, while being sensitive to changes in topology, making it a more suitable model for instance search than other techniques.

Figure 8 further details the *MAP* for different categories of

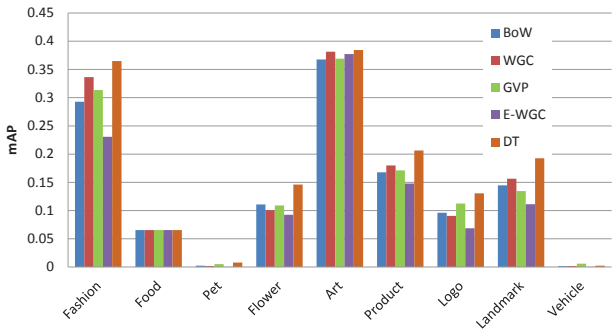

Figure 8: The performance of different approaches on various instance types.

visual instances. DT consistently shows better performances in eight out of nine categories. The instances in the "food" category mostly produce noisy BoW matches due to variations in visual appearance. All approaches can only retrieve the duplicates of a given query, resulting in equal MAP performances. For the category "pet", most failure cases are due to the excessive numbers of noisy BoW matches (e.g., the dot textures of "Dalmatian dog"). For the "vehicle" category, on the other hand, there are very few feature points extracted from the query instances (air force one, bumblebee, barrage balloon), which are texturally sparse, resulting in a poor performance. For the "landmark" category, the queries include some difficult examples, such as "Wall Street Bull" and "Bruce Lee Statue". However, DT is still able to retrieve more instances from the large dataset than other approaches. The matching for this category could become easier, if background context is also considered.

6.3 Instance Naming and Question-Answering

For ***"instance naming"***, we contrasted the performance of candidate ranking with semantic-based re-ranking (Section 4.3) turned off (PF) or on (PF+). In addition, both PF and PF+ are compared against the baseline term frequency (TF), which measures frequency by the exact matching of phrases. For fair comparison, TF is also weighted by the score from instance search, i.e., Sim^I in Equation (3). Table 3 shows the MRR@10 performance based on the result of instance search on the one million image dataset. On average, PF+ achieves a performance close to 0.5, indicating that, for most queries, the instance names can be correctly located within the top-2 positions of a ranked list. As shown in Table 3, the correct names are ranked in top-1 position for 22 out of 52 queries by PF+.

By analyzing the result, we find that, in the metadata, users tend to use dominant words or abbreviations, instead of the full instance name. For example, using "cake" instead of "Barbie cake", and "CK billboard" instead of "Calvin Klein billboard". This is especially true for phrases extracted from the description of metadata. As a consequence, TF tends to rank dominant words and abbreviations higher than the full name. PF, in contrast, by considering the syntactic relationship as in Equation (4), "cake" can also contribute to the frequency of "Barbie cake". Furthermore, by considering the potential noise from the metadata as in Equation (3), a full instance name extracted from the title can have a better chance to be upgraded to a higher rank. By further considering semantic similarity between phrases, PF+ contributes to a further improvement. For example, the rank for the instance name "leopard boot" is significantly boosted because of related words such as "leather".

Table 3: Instance naming: Comparison of MRR@10 by ranking with PF (phrase frequency), PF+ (PF plus semantics), and TF (term frequency) on the one million dataset. The () in the last row indicates the number of correct names ranked at top-1 position.

Instance	PF+	PF	TF
Fashion	0.62	0.53	0.48
Food	0.33	0.33	0.18
Pet	0.20	0.20	0.30
Flower	0.63	0.61	0.53
Art	0.43	0.36	0.57
Product	0.55	0.46	0.31
Logo	0.63	0.63	0.58
Landmark	0.67	0.50	0.33
Vehicle	0.00	0.00	0.00
Average	0.48	0.42	0.38
(top-1 #)	(22)	(17)	(15)

In general, the performance of instance naming is related to instance search. For example, the "vehicle" category yields a poor performance due to the failure in instance search. However, the relationship is not necessarily linear, and the performance is also dependent on the quality of metadata. For example, the performance of "flower" is similar to that of "fashion", though the MAP for the former is about 0.15, and the latter is more than 0.35. A special case is the "pet" category, where the performance is reasonably well, despite the low MAP in instance search. The performance is contributed by true positives from the one million distracting images, which are not labeled and considered as false responses in visual instance search.

For ***"question-answering"***, we compared strategy-1 and strategy-2 described in Section 5. The former uses the top-1 name candidate to retrieve similar questions, and then augments the parse tree for question ranking. The latter employs the top-10 candidate names for retrieval, but does not perform the name augmentation. Table 4, on the following page, summarizes the results. In terms of MRR@10, both strategies show similar performance. Basically, when a retrieved question stands out to be closely similar to an inquirer's question, both strategies are able to rank the question high. Strategy-1 shows slightly worse results, when the top-1 candidate is related but not exactly the instance name. In terms of P@10, strategy-1 shows an apparent advantage over strategy-2, by augmenting the instance name directly with "this" in the parse tree. This is because, without name augmentation, the syntactic tree matching technique mainly matches similar questions based on grammar structures. For example, the question "What is the easiest way to make this?" ("this" refers to "Barbie cake") will match to "What is the best recipe to make a tiered cake?". When the instance name is missing in the parse tree, the nouns and verbs (e.g., the words "way" and "make") are assigned with higher weights during matching. By strategy-1, the instance name is directly considered for matching, and the question "What are some tips for making a Barbie cake?" is ranked higher.

In Table 4, the category "Definition" performs much worse

Table 4: Performance for question-answering. MRR measures the first correct question found in the ranked list. P measures the proportion of correct and related questions. For the notation $X(Y)$, X is the result by considering the top-1 candidate name, while Y is by considering the top-10 candidates.

Category	*MRR@10*	*P@10*
Factoid	0.20 (0.17)	0.30 (0.21)
Definition	0.08 (0.09)	0.30 (0.20)
How-to	0.24 (0.27)	0.38 (0.31)
Opinion	0.28 (0.27)	0.35 (0.39)
Average	0.210 (0.214)	0.334 (0.279)

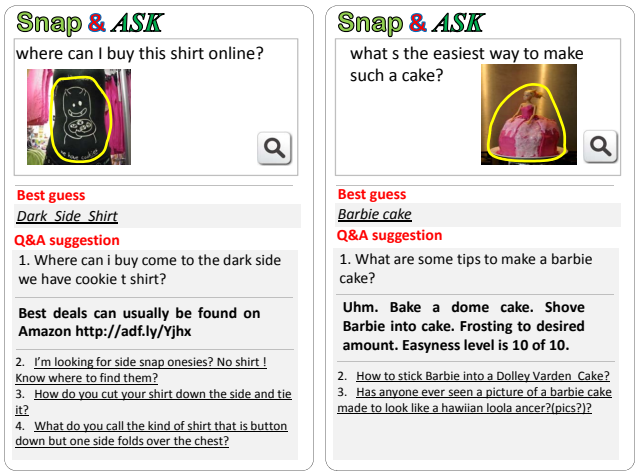

Figure 9: Examples of question-answering using our developed system. For the purpose of illustration, only the best and related questions are highlighted.

than others, mainly due to the fact that more than half of the questions fail to identify the exact instance names. However, as related, though not exact, names are found, there are still about 3 out of 10 questions related to the original questions. Figure 9 shows a few snapshots of QA examples in our testing.

6.4 Speed Efficiency

The experiments were conducted on two 8-core 2.67GHz computers with 20GB RAM each. Table 5 details the average running time for answering a multimodal question. Instance search basically completes a query within 1.5 seconds with a multi-threading implementation. The memory consumption is about 30G, mainly for keeping the Hamming signature, spatial location, dominant scale and orientation of each feature in the inverted file. Although the speed is slower than WGC, which takes about 0.9 second, the longer running time is compensated by the significantly better performance in the searching result. By processing the first 100-ranked images, instance naming takes about 11 seconds for ranking, where the time is largely spent on querying WordNet for semantic similarities. The major portion of the processing time (about 64%) is consumed by question searching and matching, when the top-1 candidate name is considered (strategy-1). Overall, the current implementation takes less than one minute to answer a question.

Table 5: Speed efficiency of our proposed framework for answering one multimodal question.

Key step	*Speed*
Instance search	1.5 sec
Name candidate ranking	11 sec
Online question search and tree parsing	5 sec
Syntactic tree matching for question ranking	18 sec
Total	35.5 sec

6.5 User Study

To evaluate the system utility, we also conducted a user study to compare Snap-and-Ask against Yahoo! Answers. A total number of 14 evaluators (8 males and 6 females) from different education backgrounds, including computer science (7), biology (2), and business (5), were invited for the subjective evaluation. The average age of the evaluators is 27, and the ages range from 24 to 31. All the evaluators are familiar with Yahoo! Answers. During the evaluation, each subject was prompted with eight different questions, of which four of them should be posted to Snap-and-Ask and the other four to Yahoo! Answers. Each question includes a text description in Chinese language referring to a picture. A subject was asked to interpret the question in English, and then pose the question to one of the systems. For Yahoo! Answers, a subject needed to textually describe the visual object for question asking. To minimize the carryover effect, questions and systems were assigned randomly. Specifically, each subject was requested to use Snap-and-Ask to answer the first four questions, followed by Yahoo! Answers for the next four questions, or vice versa.

We evaluated the systems using five criteria. At the end of each answer circle, a subject was asked to rate the system with three criteria: 1) *Accuracy*: the quality of the returned answer; 2) *Image cropping*: the use of a bounding box for question asking; 3) *Time*: the approximate time spent from formulating a question to obtaining the answer. For criterion-1, as a means to encourage full engagement in the evaluation, a subject was asked to write down the answer for each question. Note that criteria-2 is only applicable to Snap-and-Ask. For criteria-3, the evaluators were requested to note down the time they started working on a question and the time before rating the system. After completing the eight questions, a subject was also asked to compare the two systems in terms of 4) *Engagement*: how effective and efficient a system performs in returning informative answers and related questions; 5) *Acceptance*: the preference of one system over the other, based on the overall user experience. Except *Time* and *Acceptance*, all the criteria are rated on a scale of 1 to 5, with 5 being the best and 1 being the worst.

Table 6, on the following page, lists the result of the user study. Overall, Snap-and-Ask is clearly the favorite for all five criteria. Almost all subjects preferred our system when asking questions involving visual instances. From the verbal feedback, we noticed that the evaluators have a tendency to give low ratings to Yahoo! Answers, whenever they cannot name an instance and encountered difficulties in describing the visual properties. In addition, the rating of Snap-and-Ask is dependent on whether the name of the instance could be identified by the system, which in turn decides the quality of an answer. Overall, using our system, a subject spent a relatively shorter time in formulating a question, and gen-

Table 6: Comparative user study between Snap-and-Ask and Yahoo! Answers. The first three criteria show the average rating and standard deviation. The last two criteria show the preference of a system over the other, and the average time for getting an answer, respectively.

	Snap-and-Ask	*Yahoo! Answers*
Accuracy	3.6 ± 1.8	1.9 ± 1.6
Engagement	3.7 ± 1.8	1.8 ± 1.5
Image Crop	3.5 ± 0.5	*NA*
Acceptance	92.9%	7.1%
Time	2 min 20 sec	4 min 34 sec

erally regarded image cropping as a user-friendly option, especially when there are multiple objects, or when the target instance occupies a small fraction of the picture.

7. CONCLUSIONS

We have presented our framework for answering multimodal questions, along with the proposal of techniques for scalable instance search, robust naming of instance, and rigorous ranking of similar questions. The key findings include the needs for the appropriate use of spatial constraint for instance search, and syntactic as well as semantic based ranking of names and questions, for boosting the chance of finding the right answer from a number of huge but imperfect and noisy sources. For instance search, our approach, which emphasizes coherency in spatial topology while allowing local perturbation of word matching, offers apparent advantages over other methods, in terms of the scalability and robustness for non-planar and non-rigid instances occurring frequently in real-world. Experimental results demonstrate that, by searching from a million-scale image dataset and using the top-1 name candidate, an answer can be returned to an inquirer within one minute, with a good chance of finding the desired answer and three related questions out of ten suggestions.

The current system can still be improved on several aspects. Future extensions include the adaptive use of context in instance search for benefiting certain instance types (e.g., landmark), and distributed matching of similar questions by syntactic tree matching, which is currently the slowest component in our framework.

8. ACKNOWLEDGMENTS

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 118812).

9. REFERENCES

[1] J. Cao and J. F. Nunamaker. Question answering on lecture videos: a multifaceted approach. In *ACM/IEEE Conf. on Digital Libraries*, 2004.

[2] B. N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, (6):793–800, 1934.

[3] B. Erol, J. Luo, S.-F. Chang, M. Etoh, H.-W. Hon, Q. Lin, and V. Setlur. Mobile media search: Has media search finally found its perfect platform? Part II. In *ACM Multimedia*, pages 911–912, 2009.

[4] M. Gilbert, A. Acero, J. Cohen, H. Bourlard, S.-F. Chang, and M. Etoh. Mobile search in mobile devices. *IEEE Multimedia Magazine*, 2010.

[5] T. Hughes and D. Ramage. Lexical semantic relatedness with random graph walks. In *EMNLP-CoNLL 2007*, pages 581–589. ACL, 2007.

[6] M. J. Huiskes and M. S. Lew. The MIR flickr retrieval evaluation. In *ACM MIR*, 2008.

[7] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[8] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):192–212, May 2010.

[9] Y. Kalantidis, L. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis. Scalable triangulation-based logo recognition. In *ICMR*, April 2011.

[10] M. Kamvar and S. Baluja. Query suggestions for mobile search: understanding usage patterns. In *SIGCHI*, pages 1013–1016, 2008.

[11] D. Le, C. Zhu, S. Poullot, and S. Satoh. National institute of informatics, Japan at TRECVID 2011. *TRECVID*, 2011.

[12] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, Feb. 2006.

[13] G. Li, R. Hong, Y.-T. Zheng, S. Yan, and T.-S. Chua. Learning cooking techniques from youtube. In *Multimedia Modeling*, pages 713–718, 2010.

[14] G. Li, Z.-Y. Ming, R. Hong, T.-S. Chua, H. Li, and S. TangChua. Question answering over community-contributed web videos. *IEEE Multimedia*, 17:46–57, 2010.

[15] Y. Li, D. McLean, Z. Bandar, J. O'Shea, and K. A. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans. Knowl. Data Eng.*, 18(8):1138–1150, 2006.

[16] L. Nie, M. Wang, Z. Zha, G. Li, and T.-S. Chua. Multimedia answering: enriching text QA with media information. In *ACM SIGIR*, pages 695–704, 2011.

[17] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *World Wide Web Conference*, pages 161–172, 1998.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[20] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[21] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, Oct. 2003.

[22] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *ACM MIR*, 2006.

[23] K. Wang, Z. Ming, and T.-S. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *SIGIR*, pages 187–194, 2009.

[24] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma. Annotating images by mining image search results. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1919–1932, Nov. 2008.

[25] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *ACM Multimedia*, pages 218–227, 2007.

[26] Y.-C. Wu and J.-C. Yang. A robust passage retrieval algorithm for video question answering. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(10):1411 –1421, 2008.

[27] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *CVPR*, pages 25–32, Aug. 2009.

[28] H. Yang, L. Chaisorn, Y. Zhao, S. yong Neo, and T. seng Chua. VideoQA: Question answering on news video. In *ACM MM*, pages 632–641, 2003.

[29] T. Yeh, J. J. Lee, and T. Darrell. Photo-based question answering. In *ACM MM*, 2008.

[30] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry preserving visual phrases. In *CVPR*, 2011.

[31] W. Zhao, X. Wu, and C. W. Ngo. On the annotation of web videos by efficient near-duplicate search. *IEEE Trans. on Multimedia*, 12(5):448–461, 2010.

[32] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *ACM Multimedia*, 2010.