ABSTRACT

Images are often used to convey many different concepts or illustrate many different stories. We propose an algorithm to mine multiple diverse, relevant, and interesting text snippets for images on the web. Our algorithm scales to all images on the web. For each image, all webpages that contain it are considered. The top-K text snippet selection problem is posed as combinatorial subset selection with the goal of choosing an optimal set of snippets that maximizes a combination of relevancy, interestingness, and diversity. The relevancy and interestingness are scored by machine learned models. Our algorithm is run at scale on the entire image index of a major search engine resulting in the construction of a database of images with their corresponding text snippets. We validate the quality of the database through a large-scale comparative study. We showcase the utility of the database through two web-scale applications: (a) augmentation of images on the web as webpages are browsed and (b) an image browsing experience (similar in spirit to web browsing) that is enabled by interconnecting semantically related images (which may not be visually related) through shared concepts in their corresponding text snippets.

Categories and Subject Descriptors

H.2.8g [Information Systems]: Database Management: Database Applications: Image Databases

General Terms

Data Mining Algorithms, Applications

Keywords

Text mining for images; Text snippets; Interestingness; Relevance; Diversity; Browsing; Semantic image browsing; Web image augmentation

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD ’14 August 24–27, 2014, New York, NY, USA
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623346.

1. INTRODUCTION

The popular adage “a picture is worth a thousand words” reflects the effectiveness of an image in conveying many different ideas or “stories.” An image is typically used to illustrate one of these stories in any given webpage. However, the same image can be used in different webpages to illustrate different stories at different levels of descriptive and interestingness. By collecting the stories associated with an image together, we can create a host of new applications that seamlessly integrate the image and text modalities.

The focus of our paper is exactly this: We propose a mining algorithm to obtain the most relevant and interesting text snippets for images on the web. Using this database of images and their associated text snippets, we present two new applications that we have implemented at web scale. We believe, however, that these applications represent the “tip of the iceberg”: many more applications are possible.

Figure 1 shows a sample of results for a few images. The images are selected to cover many of the important types of images on the web; people, travel, music, etc. For each image, we show two of the many snippets identified for it. The snippets are related to the image in the most general sense: memories or events behind the image, an analogy to establish context, or even descriptions of a historical event in the context of the image. Note, however, that the text generally goes far beyond a simple description of the visual contents of the image. For example, a visual description of Figure 1(b) might be “someone snorkeling in the Maldives.” Instead, our mined snippets include interesting information about the monsoon season and the geography of the islands, information that is not apparent in the image itself.

Our first contribution is to propose a scalable solution to mine the web for multiple relevant, interesting and diverse textual snippets about the image. The scale of our solution has two dimensions: we consider all images on the web and for each image we consider all web pages that contain the corresponding image. This scale enables reliable identification of multiple high quality snippets for images that are discussed widely on the web. Note that our approach differs considerably from prior work in understanding images [7, 13, 12, 16, 20, 22, 28], where the goal is to synthesize textual descriptions of the visual content of the image by recognizing objects, their attributes, and possibly using similar images in image databases to borrow captions.

With the web as a repository of world knowledge, we operationalize our solution based on a key insight: if an image is interesting, multiple people will post it in webpages, including stories related to it. These stories will vary in their
Figure 1: Examples of text snippets found by our algorithm. We just show the top two snippets, although generally many more are found. Note that the snippets are not purely descriptions of the content of the image. Instead, they contain all manner of interesting facts and stories related to the content.

2. RELATED WORK

There are two main bodies of related work. The first set focuses on analysing an image to generate a text description or caption. The second body of work is that of document (i.e. webpage) summarization. We discuss each in turn.

Image caption generation: Possibly the earliest work in this area is [20], which focused on associating word tags with an image by analyzing image regions at multiple granularity. Subsequently, in [12, 16, 28] the semantics of the image pixels are inferred to generate image descriptions. In particular, [12] detects objects, attributes and prepositional relationships in the image. These detections are composed to generate descriptions of the image. In [28], the image is decomposed at multiple granularity to infer the objects and attributes. The outputs of this step are then converted into a semantic ontology based representation that feeds into a text generation engine. A related line of work poses caption generation as a matching problem between the input image and a large database of images with captions [7, 13, 22]. In [7], a learned intermediate semantic space between image features and the tokens in the captions are used to generate sentences for images. In comparison, [22] constructs a data set of 1 million images spanning over 87 classes. Given an input image, this approach transfers captions from the closest matching image in the data set. In [13], phrases extracted from the caption of the matched image are used to generate a descriptive sentence. The goal of all these techniques is to describe the image content while we would like to capture the stories conveyed in the context of the image, and is not restricted by what is contained in the image.
Document Summarization: There is a large body of work in automatic text summarization. See [15, 23] for surveys. In this literature, the goal is to summarize documents, either by identifying key phrases and sentences that are reflective of the focus of the document (extractive approaches) [9, 4, 5] or by paraphrasing the content (abstractive approaches) [3]. In [9], sentences are extracted from one or more articles about the same topic. In [4] key sentences from webpages are extracted for viewing on a mobile device. In another example, [3] summarizes a webpage using a probabilistic model of the gist of the webpage. The underlying goal of all these methods is to identify the central focus of the document to be summarized. In contrast, we would like to identify content related to the image. One can think of using these techniques if we know the region of text that corresponds to image, which is a hard task by itself [2].

3. SNIPPET MINING ALGORITHM

Our algorithm is based on the insight that if an image is interesting, multiple people will embed it in a webpage and write about it. For each image (along with near-duplicates), one can mine the web for all the webpages containing it in order to identify text snippets that are relevant and interesting and also form a diverse set of text.

With this insight, we have developed a highly scalable solution to mine the web for the interesting text snippets related to an image. Figure 2 provides the overview. The web images are clustered into “near duplicate” groups (§3.1), referred to as “duplicate image set” or “image set,” for short. While each image set corresponds to a single image, these images reside in multiple webpages, PURL (page URL in which the image resides) with a unique media url (MURL). Hence, each image set is represented by a collection of triplets \{MURL, PURL, HTML\} where HTML is the associated HTML source of PURL. Using this representation, we identify candidate snippets for each image set (§3.2). Then, the top K snippets that are most diverse, interesting and relevant to the image (§3.3) are identified by posing the problem as the combinatorial optimization problem of subset selection. We use machine-learned models of relevance and interestingness to score the snippets which are used within the optimization framework (§3.3.1).

3.1 Scalable Image Set Identification

We would like to cluster images such that each cluster consists of images that are near duplicate to each other. Images within a near duplicate image set can differ in their sizes, compression ratios, cropping, color transformations and other minor edits. For this purpose, we require a clustering algorithm that covers (a) large variation within a duplicate image cluster while minimizing false positives and (b) is highly scalable for clustering billions of images on the web.

To satisfy both these requirements, we use two-step clustering method [26] that is scalably implemented using hashing techniques within MapReduce framework [6]. The technique combines global and local features, with global descriptors to discover seed clusters with high precision and the local descriptors to grow the seeds to obtain good recall.

For the global descriptors, we first partition each image into $8 \times 8$ blocks and compute the mean gray value for each block. The image is then divided further into $2 \times 2$ blocks, and a histogram of edge distributions in evenly-divided 12 directions is extracted from each block, plus one dimension of non-edge pixel ratio. The histograms of mean gray values and edge distributions are cascaded into a 116-dimensional raw feature vector. A pre-learnt Principle Component Analysis (PCA) model is then applied and the first 24 dimensions of the PCA-ed vector are retained. The local descriptor of an image is a truncated set of visual words defined on a visual codebook of 1 million codewords. The local features proposed in [27] are first extracted and then ranked by their average Difference-of-Gaussian (DoG) responses. The top-ranked 40 features are used as the local descriptors of an image. Two images are near duplicates if they have 6 visual words in common.

Seed clusters are generated in three steps. First, each image is encoded into a 24-bit hash signature by quantizing each dimension of the global descriptor against the PCA mean. These signatures then serve as keys to divide the images in a MapReduce framework (i.e. the Map step). Second, images with identical hash signatures are assigned to the same hash buckets. Each bucket is processed within one vertex of the MapReduce machine cluster (i.e. the Reduce step). Third, pair-wise distance matching is performed within a bucket.

We grow the seed clusters by merging them using local descriptors. For computational tractability, we assume that near duplicate images reside in neighboring buckets in the global feature space, where neighboring buckets are those whose corresponding signatures are different in at most two bits. We randomly select one image from each seed cluster and add to the cluster all the images from neighboring buckets that have 6 or more visual words in common.

3.2 Forming Candidate Snippets

Each duplicate image set is represented by triplets \{MURL, PURL, HTML\} of the image url, page url and the HTML source of the page. Along with this, we parse HTML to obtain a linear ordering of the text and image nodes and this is maintained as $W^{\text{PURL}}$.

Corresponding to each text node $s^{\text{PURL}}_n$ in $W^{\text{PURL}}$, we generate a candidate snippet $< s_{n,\text{PURL}}, p_{n,\text{PURL}} >$, where $s_{n,\text{PURL}}$ is the $n^{\text{th}}$ PURLth snippet text and $p_{n,\text{PURL}}$ is its location in $W^{\text{PURL}}$. The set of candidate snippets generated is

\begin{itemize}
  \item \textit{We restrict triplets to be from English-language non-spam websites. We also exclude social networking and photo sharing sites that do not contain enough textual content.}
  \item \textit{Non-English and/or ill-formed sentences are ignored.}
\end{itemize}
\[
\{ < \textit{sn}^n, \textit{purl}^n, \textit{m}^n, \textit{purl}^m > \}_{n=1}^{N^\textit{purl}} \] where \(N^\textit{purl}\) is the number of snippets.

Similarly, we represent an image as follows: For each image node corresponding to the \(\textit{MURL}\), we extract its associated “Alt” text or “Src” text. We denote this text \(< \textit{m}^\textit{purl}, \textit{L}^\textit{purl}>\), where \(\textit{m}^\textit{purl}\) is the text and \(\textit{L}^\textit{purl}\) is the location of the image node in \(W^\textit{purl}\).

### 3.3 Top K Snippet Selection

For each image, \(I\), let \(S\) be the candidate set of snippets identified from the set of triples \{\(\textit{MURL}, \textit{PURL}, \textit{HTML}\)\} (§3.2). From this large candidate set, we would like to select \(K\) top snippets, \(T \subset S\) which are not only relevant and interesting to \(I\) but also exhibit diversity. This selection can be guided by the objective function

\[
F(T|I) = \lambda \sum_{s \in T} \text{rel}(s) \text{int}(s) + (1 - \lambda) H_0(T),
\]

that trades-off total gain given by the sum of the product of relevance \(\text{rel}(s)\) and interestingness \(\text{int}(s)\) of each snippet with their diversity as measured by the entropy of the set \(H_0(T)\) (larger the entropy, larger the diversity).

In our objective (eq. 1), we consider the product \(\text{rel}(s) \text{int}(s)\) instead of sum \(\text{rel}(s) + \text{int}(s)\) because we prefer snippets that are both simultaneously relevant and interesting. We also differentiate between relevance and interestingness and not use a single term since a highly relevant snippet may not be very interesting, and vice versa (See Figure 4).

The optimal subset satisfies:

\[
T^* = \arg \max_{T \subset \cup_{T | |T|=k}} F(T|I).
\]

While one can solve for the optimal subset through exhaustive search over all possible subsets of \(S\), this is clearly prohibitive even for a reasonable size of \(S\).

However, for suitable choices of \(\text{rel}(s)\) and \(H_0(T)\), Eq. 1 is submodular. That is, it exhibits the property of “diminishing returns” so that the difference in the value of the function that a single element makes when added to an input set decreases as the size of the input set increases. Mathematically, if \(X \subset Y \subset S\), then adding an element \(z \in S\) to both \(X\) and \(Y\) should satisfy:

\[
F(X \cup z) - F(X) \geq F(Y \cup z) - F(Y)
\]

As long as we choose \(\text{rel}(s)\) \(\geq 0\) so that \(\sum_{s \in T} \text{rel}(s)\) is a monotonically increasing function, Eq. 1 will be submodular since entropy is submodular [11]. We describe one such choice for \(\text{rel}(s)\) in §3.3.1.

The advantage of submodular functions is that while being computationally difficult, we can design greedy solutions that are at most \((1 - \frac{1}{e})\) times worse than the optimal solution [21]. Using this theoretical guarantee, we design a greedy solution as follows: Initialize with a snippet that has the largest value of \(\text{rel}(s)\). Then, iteratively add snippets, one at a time, such that the added snippet (along with previously chosen set of snippets) maximizes Eq. 1.

#### 3.3.1 Relevance and Interestingness

The top-K snippet selection formulation (Eq. 1) uses separate functions to score a snippet for relevancy \(\text{rel}(s)\) and interestingness \(\text{int}(s)\) to the corresponding image. Here, we describe the instantiation that we use in this paper for these two components. We used separate machine learned models of regularized linear regression.

The regularized linear regressor captures the relationship between features, \(\phi(x)\), extracted from the snippet \(x\) and its scores through the functional form \(y = w^T \phi(x)\). The details of the features used are provided momentarily. Given some training data \(x_i \in X\) that has been annotated with the relevance (or interestingness) scores \(y_i\), the corresponding parameters \(w\) are learned by solving the optimization function:

\[
w^* = \arg \min_w \lambda ||w||^2 + \sum (w^T \phi(x_i) - y_i)^2
\]

where \(\lambda\) is a regularization parameter. The unique optimum of Eq. 4 can be found via gradient descent [14].

#### Features used:

First, we describe the vocabulary for representation:

Let \(\mathcal{M} = \cup_{\textit{purl}} \{\textit{m}^\textit{purl} \cup \cup_{n} \textit{sn}^n, \textit{purl}^n\}\) be the union of the image representation and all the candidate snippets for the image set. Let \(\mathcal{V}\) be the unigrams (sans stop words) identified in \(\mathcal{M}\).

Each snippet \(s\) is represented using a \(|\mathcal{V}|\) vector, \(\mathbf{s}\), such that \(s[k]\) is the number of times that the \(k\)th unigram in \(\mathcal{V}\) occurs in \(s\). Similarly, for image \(I\), \(\mathbf{m}\) is a \(|\mathcal{V}|\) vector such that \(m[k]\) is the sum of the number of times that the \(k\)th unigram in \(\mathcal{V}\) occurs in the union of the representations of the images across all \(\textit{PURL}\) in \(\cup_{\textit{purl}} \{\textit{m}^\textit{purl}\}\).

The feature vector, \(\phi(\cdot)\), consists of nine features from the following five groups of features:

- **Match Score**: While \(s\) corresponds to a single snippet, \(\mathbf{m}\) is an aggregate over all image representations (captions, Alt Text) across multiple instances of the image. Therefore, we measure similarity between \(s\) and \(\mathbf{m}\) as a dot product between \(s\) and normalized \(\mathbf{m}\):

  \[
  \text{matchscore}(s, I) = \mathbf{s}^T \frac{\mathbf{m}}{||\mathbf{m}||}
  \]

- **Context Scores**: When the text around a snippet is relevant, then this specific snippet is likely to be relevant too. With this intuition, we compute two features which are average of the Match Score on windows around the snippet in question. Let \(\rho_{j,n,\textit{purl}} = \delta[\textit{sn}^n, \textit{purl}^n - K, \textit{sn}^n, \textit{purl}^n + K]|\textit{purl}^n\) be a indicator function which evaluates to 1 if \(\textit{sn}^n, \textit{purl}^n\) is in the range \([\textit{sn}^n, \textit{purl}^n - K, \textit{sn}^n, \textit{purl}^n + K]\). The context scores are then given by:

  \[
  \text{contextScore}(\textit{sn}^n, \textit{purl}^n, I) = \frac{\sum_{j=1}^{N^\textit{purl}} \rho_{j,n,\textit{purl}} \text{matchscore}(\textit{sn}^n, \textit{purl}^n, I)}{\sum_{j=1}^{N^\textit{purl}} \rho_{j,n,\textit{purl}}}
  \]

In our experiments, we used two window sizes, corresponding to \(K = 5\) and \(K = 30\).

- **HTML Parse Distance**: Good snippets tend to be closer to the image location. Therefore, we also capture distance from the snippet to the image node as:

  \[
  \text{ParseDist}(\textit{sn}^n, \textit{purl}^n, I) = \textit{ln}^n - L^\textit{purl}
  \]
Measure of SPAM: When a snippet contains a lot of repeated words, it is less likely to be relevant or interesting. We use entropy of the words present to capture this idea: spamminess(Sn,PURL) =

\[- \sum_{k, Sn,PURL(k)>0} \frac{Sn,PURL[k]}{\text{PURL}_k} \log \left[ \frac{Sn,PURL[k]}{\sum_r Sn,PURL[r]} \right]. \tag{8}\]

Linguistic Features: The interestingness of a sentence often depends on its linguistic structure. We use four linguistic features: (1) the length of the sentence with the intuition that longer sentences are more interesting (2) whether the sentence begins demonstrative (such as beginning with “this” or “these”) (3) whether the sentence is first person, beginning with “I” or “we” and (4) whether the sentence is definitional, i.e., begins with a pronoun and then includes the word “is” or “are” afterwards.

Data set for learning
To learn these functions, we construct a training set as follows: We randomly chose 250 images. For each image, we assembled all the snippets that are within reasonable distance to the MURL in the HTML parse tree (so that clearly irrelevant snippets are not considered). This resulted in 5443 image-snippet pairs. For each pair, we obtain human judgments independently for relevance and interestingness.

Human judgments: We designed a Human Intelligence Task (HIT) on Amazon Mechanical Turk to label each pair of image and snippet. Figure 3 shows the HIT setup for two example pairs. Given this HIT, the judges were asked to independently score the snippet for relevance and interestingness on a discrete scale between zero and three, where zero means not relevant (not interesting) and three corresponds to very relevant (very interesting). Each pair was judged by approximately 10 judges. The average relevance of the snippets is 1.57 (about half way between “Somewhat Relevant” and “Relevant”). The average interestingness score is 1.17 (a little bit higher than “Somewhat Interesting.”).

Figure 3: HIT used to collect training data to learn the snippet scoring functions. We showed image-snippet pairs to the judges and asked them to rate the snippet based on its “Relevance” to the image and on how “Interesting” it is. Some sentences are highly relevant, but not very interesting (bottom).

Figure 4: An example image with three snippets and their average human labels. (a) illustrates a snippet that is relevant but not very interesting. (b) is much more interesting. (c) is more interesting than relevant; it is about the author, not his character.

In Figure 4 we include an image to illustrate the human labeling. We include three snippets together with the human scores. Snippet (a) is provided as an example of why it is important to assess the interestingness of the snippets. Snippet (a) is a fairly factual statement about the image. It is scored as relevant, but as not particularly interesting. In comparison, (b) is significantly more interesting. Snippet (c) is notable in that it has a relatively low relevance (compared to the interestingness, which is generally scored more conservatively than relevance.) The sentence is about “Dr. Seuss” whereas the image is one of his characters.

3.3.2 Entropy of Set:
The normalized entropy, H0(T) in Eq. 1 models the diversity of the chosen sentences. We would like to favor sentences that are diverse in two respects. First, they differ in their vocabulary usage. Second, they are derived from different webpages. Hence, the entropy H(T) of set T is computed over N+1 binary random variables, where N of them constitute the vocabulary used in snippets in T and the (N+1)th variable models the webpage variability among T. The probability of the n-th variable to have a value of 1 is given by the fraction of times the value is observed in the set T. The normalized entropy is given by H0(T) = H(T)/log(N + 1) which normalizes the entropy across different subsets. As we add snippets, the incremental gain in entropy will diminish due to the fact that the vocabulary is fixed.

3.4 Scaling to the Web
We have run our snippet mining algorithm on the entire web image index of a major search engine with billions of images. We provide details of scaling up the image clustering along with the details of the clustering in (§ 3.1). After the clustering step is performed, the algorithm is embarrassingly parallel and is performed independently for each image set. In fact, on a 100 core cluster, the entire processing took about 45 hours. We can also run the pipeline for delta updates to introduce clusters of new images or additional snippets for the same image set.
4. EVALUATION OF SNIPPET MINING

We now provide an experimental validation of our algorithm using a quantitative comparison with two reasonable baselines (§ 4.1) and quantitative coverage results on popular web queries for image search retrieval (§ 4.2).

4.1 Comparison with Baselines

To the best of our knowledge, there is no prior work on extracting a set of text snippets for an image on the web. There are, however, two reasonable baselines:

- **Query-by-Image and Webpage Summarization (QbI/WS):** Find all occurrences of the image on the web and then use a webpage summarization algorithm to generate a snippet for each page. We compare with a commercial system (http://images.google.com) that follows this approach for a subset of its results. In particular, we use the snippets shown for the top three results from the “Pages that include matching images.”

- **Im2Text using Visual Features:** We also compare with the approach of Ordonez, Kulkarni, and Berg [22]. This algorithm matches the image to a database of 1 million Flickr images that already have captions, and transfers the captions from the best matches.

**Data set:** Web image search retrieval algorithm such as QbI/WS are most tuned for head (popular) images and their landing pages. Therefore, images that are most popular on web image search will serve as better baselines. We take the top 10,000 textual queries to a popular search engine and randomly selected 50 images that are among the top ranking results for those queries as the test set.

The results for two images are included in Figure 5. We found that the QbI/WS algorithm always found the correct image and generated plausible results. The results using purely visual features for comparison [22] are far less relevant. This algorithm never really found the correct topic of the image, due to a mismatch between their database (1 million Flickr images on narrow topics of interest) and the web (billions of images). We therefore dropped the Im2Text algorithm [22] from the subsequent quantitative comparison.

**Mechanical Turk Setup:** In order to perform a quantitative evaluation of our approach with QbI/WS, we conducted pairwise evaluations focused on the overall preference of one ranked list over other. We used Amazon Mechanical Turk with each Human Intelligence Test (HIT) consisting of a pair of ranked lists of snippets, corresponding to our proposed algorithm and QbI/WS. To remove the presentation bias, we considered both orderings of the lists. Each ordering is a separate HIT. We also had each HIT judged by five judges. Each judge was asked to study the two lists and specify which of the two they preferred. They also had the option of choosing “both are comparable”. The judges were asked to consider multiple dimensions while making their judgements: relevancy, interestingness, diversity and repetitiveness of content. They were required to spend at least five minutes on the task (we discarded answers from judges that did not conform to this guideline).

**Results:** Figure 5(bottom) shows the key results. Of all the judgments, 72% exclusively preferred our approach.

We used the software available at http://vision.cs.stonybrook.edu/~vicente/subcaptions/

---

Our Results:

- Banana is the common name for any of the very large, tree-like, herbaceous plants comprising the genus Musa of the flowering plant family Musaceae, characterized by an above-ground pseudostem (false stem) with a terminal crown of large leaves, and hanging clusters of edible, elongated fruit.
- These bananas, grown right along side the railroad tracks, would benefit him and the United Fruit Company in the future when the railroad would help transport the bananas to the United States (Landmeier).
- Heartburn: Bananas have a natural antacid effect in the body, so if you suffer from heartburn, try eating a banana for soothing relief.

---

Ordonez, Kulkarni, and Berg:

- I put extra challenge on still life photography by taking on a white teapot and cup on a white background. It was fun!
- A broken sphere plastic bull under two regular tungsten bulbs
- This would be boy 6 in a row! Boy Fever in the Tanner house hold :)

---

Preferences:

Ours: 100%
Tie: 0%
QbI/WS: 0%

---

Figure 5: Top and Middle: Qualitative results from a comparison between our algorithm and two baselines. The first baseline is a Query-by-Image/Webpage Summarization algorithm that finds the image on the web and then summarizes each webpage independently. The second baseline is the algorithm of Ordonez, Kulkarni, and Berg [22]. Bottom: A quantitative comparison between our algorithm and the Query-by-Image/Webpage Summarization algorithm. We took the results for 50 images and asked judges which they preferred. No preference was also an option. Bottom Left: 72% of all votes preferred our results, compared to just 10% for the Query-by-Image/Webpage Summarization algorithm. Bottom Right: When we tally up the votes for each image, we found that the judges preferred our results for all 100 images.
Figure 6: Results on popular queries to an image search engine. We took the top 10,000 queries and considered the top 50 result images for each query. We plot the number of result images with at least $K = 3, 4, 5$ text snippets (X-axis) against the page-view weighted percentage of queries (Y-axis) that have at least that number of results images with K text snippets. The results show that most queries have 10-30 results images with snippets.

17.8% felt “Both are comparable,” while 10.2% preferred the QbI/WS algorithm. We also looked at the individual results. For all images, a (strictly) greater number of judges preferred our approach to the QbI/WS algorithm.

The qualitative results in Figure 5 were chosen to give one example (top) where our results are unanimously preferred, and one example (middle) where a relatively high percentage of judges prefer the Query-by-Image/Webpage Summarization results (20% compared to the average of 10.2%).

4.2 Coverage for Popular Images

Although we ran our algorithm on the entire web index of a major search engine, we do not obtain results for all images. We impose a threshold on the overall score $rel(s) \cdot int(s)$ to remove poor quality results. We now present a set of experimental results to illustrate how likely a snippet will be available for the most popular images; i.e. the ones that are returned as results for the most common queries to Bing image search engine.

Experiment Setup: We consider the top 10,000 most frequent queries issued to Bing image search engine. For each query, we considered the top 50 image results, corresponding to the first page of image search results. We used our approach to identify the top $K$ snippets. Some of these results may not have $K$ snippets associated with them. Next, we compute a metric that captures this property.

Metric: We use the percentage of queries weighted by the frequency with which they were issued. Each query in the top 10,000 was issued by multiple users over the one month sampling interval. When computing the percentage of queries that meet a criterion, we weight by the number of times the query was issued. We refer to this metric (perhaps slightly confusingly) as “page-view weighted.”

Results: The results are plotted in Figure 6. On the X-axis, we plot the number of result images that have a least $K = 3, 4, 5$ text snippets. On the Y-axis we plot the page-view weighted percentage of queries that have at least that many (X-axis) results images with K text snippets. The results in Figure 6 show that for essentially all queries, there are some results images with text snippets. Most queries have 10-30 results images with text snippets. We qualitatively investigated the nature of the images for which no high quality text snippets were found. Example of the sort of images are shown in Figure 7. (a) Personal photos that are posted to blogs and photo-sharing cites rarely have high quality text associated with them, even if they are subsequently reposted multiple times. (b) As we currently only process English language webpages, foreign language images do not result in high quality snippets. (c) Commercial images are often repeated across multiple webpages with very similar text and are removed as having too much text duplication. (d) Generic icons can rarely be associated with high quality text.

5. APPLICATIONS

The database of text snippets mined by our algorithm has a variety of possible applications. The snippets could be shown to users along with image search results. They can be used to improve image search relevance. It may also be possible to use the snippets to discriminate more interesting images from less interesting ones. Images for which there are a lot of interesting snippets (see Figure 1) are generally far more interesting than images for which not much text is found (see Figure 7).

We now propose two other applications. The first consists of augmenting images on the web when a user views the page (§ 5.1). The second is an image browsing application built by connecting images through their snippets (§ 5.2).

We investigated the queries for which there are no results with text snippets. These were mainly cases where the top 50 results images are classified as adult content, but the query was not. We restricted our processing to non-adult images and so obtained no results in these cases.
Figure 8: A screenshot of the web augmentation application (§ 5.1). The plug-in is part of the Bing Bar toolbar at the top. The text results are populated over the image as text pop-ups that show up on mouse-over. Please see http://research.microsoft.com/en-us/um/people/sbaker/1/www/augmentation.mp4 for a video.

These applications both follow a client-server architecture. For both application, we built a Windows Azure [19] service that hosts the database of text snippets extracted using our algorithm. This service is used by the client to query for snippets corresponding to an image and/or for images pertaining to key words that appear in a text snippet.

5.1 Web Image Augmentation

Often when an image is included in a web page, only a brief caption is provided. Enriching these images with additional interesting information can improve the web browsing experience for the user. If the information comes with associated links to its source, the user can easily choose to explore the additional information by clicking on the links. To this end, we developed a ‘Bing Bar’ plug-in for Internet Explorer [18] that automatically identifies all the images on the current web page, and augments them with any snippets that were found by our algorithm.

Figure 8 shows a screenshot of the application. After the webpage has been rendered, the HTML is passed to the plug-in. The plug-in parses the HTML, extracts the URLs of the images, and queries the Windows Azure service to obtain the snippets for the images. The plug-in then injects JavaScript into the HTML that renders the text snippets when the user hovers over the image and then clicks on one of the boxes that are displayed. The plug-in then passes the HTML back to the browser which re-renders the page. Figure 8 shows such an example snippet for an ‘Adele’ image. For a detailed demo, please see the video http://research.microsoft.com/en-us/um/people/sbaker/1/www/augmentation.mp4.

There are several advantages of this system over other interfaces for viewing query-by-image results. First, the browsing experience is completely passive. The user does not need to initiate a query for every image. The query is initiated automatically, and only when there are interesting results, are they displayed in an un-obtrusive manner. Secondly, this application adds virtual hyperlink on top of the Web, from images to related content on other web pages. This can enrich the browsing experience for the user.

5.2 Semantic Image Browsing

A number of recent works have focused on building effective visualizations of a collection of images that are retrieved in response to a query of interest [10, 24]. In this section, we present a system for a never-ending browsing experience that revolves around images and the text snippets associated with them (found using our algorithm). The starting point for the application is either an image or a concept of interest, which can be user-provided or chosen at random. The choice of the next image is based on the text snippets, thereby allowing transitions between images that can be visually unrelated, but semantically related. In particular, we identify concepts in the snippets which are then used as the means for the user to control the choice of the next image in a semantically meaningful manner. See Figure 9.

Browse interface: The interface allows people to browse images on the web. As they browse, the system displays a set of text snippets extracted using our algorithm. These snippets provide interesting information about the current image being viewed. We also detect concepts phrases in the snippet that map to Wikipedia article titles [8, 17, 25], further refined using the techniques proposed in [1]. These concept phrases are used to hyperlink to other images that share the concept phrases in their snippets. Our algorithm for determining the destination of the hyperlink is randomized; if the viewer returns to the same image twice, we want them to explore a different path the second time around. Using an inverted index, we locate K=10 top images at random that have a text snippet that includes the concept being clicked on. There are typically thousands of possible destinations. We then sort these images by the frequency with which the concept appears across all the snippets associated with the image, and choose the image with the highest frequency. We found this algorithm yielded a compelling experience. More sophisticated approaches are left as future work.

Figure 9 contains a screenshot of our application. The current image is a diagram explaining how neural signals are transmitted from an axon to a dendrite in the brain. The displayed text snippet explains the process. The history of recent images is shown on the left side. We began with an image of Ernest Rutherford, then tapped on Isaac Newton, then Albert Einstein twice, once a photo of Albert and his wife, the second time just the famous physicist. We then tapped on brain and neuron before arriving at the current image. The right side of the display includes a list of all the concepts for this subset of science images. The user can tap on any concept to initiate a browsing session at a different starting point. An illustration of the browsing session in Figure 9 is contained in the web video http://research.microsoft.com/en-us/um/people/sbaker/1/www/browsing.mp4.

Evaluation using Mechanical Turk: We evaluated the effectiveness in identifying the next image to browse, based on the semantics conveyed by the snippets. In particular, are two images that are visually unrelated, indeed related through the snippets that describe them? To answer this question, we make use of an evaluation set consisting of 500 pairs, <x,y>, of images (along with the snippets that link x to y) constructed as follows: for each pair, we choose an image x at random. Then, we choose a concept associated with x (from the snippets of x) at random. We then choose the next image y, using the algorithm described above to compute the destination of a link within our application.
We conduct two sets of experiments using Amazon Mechanical Turk. In the first experiment, each Human Intelligence Task (HIT) consists of only the pair of images, \( <x_i, y_i> \). Each judge was asked to specify if they found the pair of images to be ‘related’, ‘not related’ or ‘not sure’. In the second experiment (a different HIT), the judges repeated the same experiment, but now, in addition to the pairs of images, they were also shown snippets linking the two images with the connecting concept highlighted.

**Results:** Figure 10 shows the average number of images that were judged in each of the categories. We can see a huge drop in fraction of ‘unrelated’ images from 57% to 25% showing the efficacy of our approach in linking semantically related images that are seemingly unrelated. As examples, consider the pair of images shown in Figure 10. In isolation, the connection between the two images is not obvious. With the addition of the text snippet, the viewer immediately finds both images more interesting and sees the ‘semantic’ relationship between them.

### 6. CONCLUSION

We have presented a scalable mining algorithm to obtain a set of text snippets for images on the web. The optimal set of snippets is chosen to maximize a combination of relevance, interestingness, and diversity. The relevancy and interestingness are scored using learned models.

There are a number of applications of the resulting text snippets. One possibility is to display the snippets along with image search results. In this paper we proposed two others, a plugin to an internet browser that augments web-
pages with snippets overlaid on images, and a tablet application that allows users to browse images on the web via hyperlinks embedded in the text snippets. The snippet data can be useful for improving image search relevance.

One suggestion for future work is to analyze the snippets in more detail, for example by clustering, to find groups of related images. The results could be used to broaden the set of snippets and concepts associated with an image, possibly leading to deeper understanding of the content of the images, and more interesting browsing experiences.

7. ADDITIONAL AUTHORS
Rizwan Ansary (Microsoft, rizwan@microsoft.com)
Ashish Kapoor (Microsoft, akapoor@microsoft.com)
Qifa Ke (Microsoft, qke@microsoft.com)
Matt Uyttendaele (Microsoft, mattu@microsoft.com)
Xin-Jing Wang (Microsoft, xywang@microsoft.com)
Lei Zhang (Microsoft, leizhang@microsoft.com)

8. REFERENCES