

# Clustering Image Search Results by Entity Disambiguation

Kaiqi Zhao, Zhiyuan Cai, Qingyu Sui, Enxun Wei, and Kenny Q. Zhu

Department of Computer Science & Engineering  
Shanghai Jiao Tong University, China  
{kaiqi\_zhao, luckyvega}@163.com,  
{sqybilly, weienxun}@gmail.com, kzhu@cs.sjtu.edu.cn\*\*

**Abstract.** Existing key-word based image search engines return images whose title or immediate surrounding text contains the search term as a keyword. When the search term is ambiguous and means different things, the results often come in a mixed bag of different entities. This paper proposes a novel framework that understands the context and thus infers the most likely entity in the given image by disambiguating the terms in the context into the corresponding concepts from external knowledge in a process called conceptualization. The images can subsequently be clustered by the most likely associated entities. This approach outperforms the best competing image clustering techniques by 29.2% in NMI score. In addition, the framework automatically annotates each cluster of images by its key entities which allows users to quickly identify the images they want.

## 1 Introduction

Images are one of the most abundant multimedia resources on the Web. Most commercial search engines offer image search today, which enables the user to retrieve images by search terms. By default, all existing image search engines rank the returned images by the relevance of their contexts (i.e. the web pages they are embedded in) to the query keywords. Fig. 1 shows the result for searching “bean” on *Google Image* in October 2013. The result appears to be a random mix of many different entities related to the keyword “bean”, e.g., “Mr. Bean (comedian)”, “Sean Bean (actor)”, “beans (crop)”, etc. Ambiguous search terms like this are not rare: Google Image returns at least two different entities for “kiwi”, three for “explorer”, and over ten different persons named “Jerry Hobbs”!

This paper is concerned with the problem of clustering web images according to the entity or concept they represent. Once the images are clustered, the search engine can return the *original* set of search results classified by distinct entities, offering easier accessibility and more diversity. Note that a separate but different problem [21,22] is mapping images to an entity in a knowledge base like Wikipedia or YAGO [20]. That is a different problem because 1) the entity is unique and known in advance, so its features in the knowledge base can be used for retrieving images whereas our problem does not

---

\*\* Kenny Q. Zhu is the contact author and is supported by NSFC Grants 61100050, 61033002, 61373031 and Google Faculty Research Award.

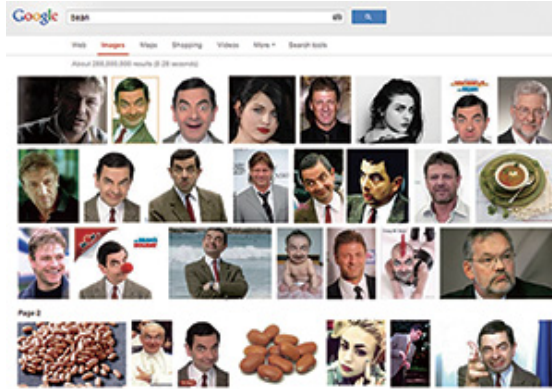


Fig. 1. Search Result of “bean” on Google Image

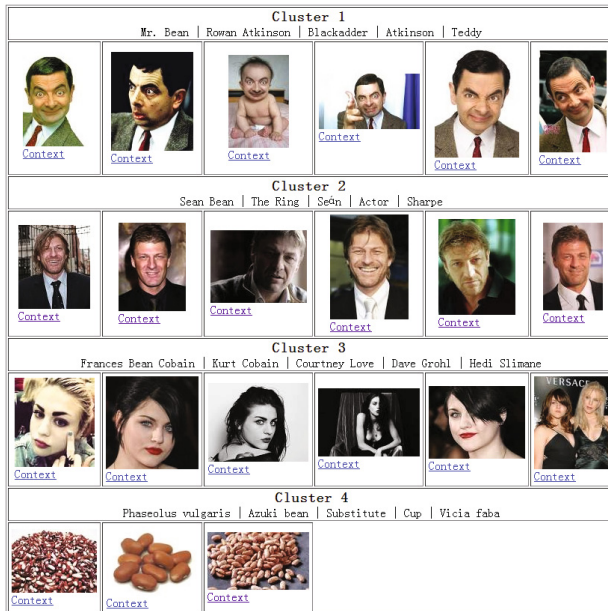
assume known entities *a priori*; 2) the goal is to rank the relevant images to an entity while our problem is a clustering problem.

In the past, there have been numerous research efforts on image clustering. These efforts can be roughly divided into three categories: visual-based, context-based and hybrid approaches.

*Visual-based methods* only take into account visual features such as SIFT descriptors, edge histogram, color and contrast [11,27], and these are often insufficient for distinguishing real entities. For example, some images of *Mr. Bean* in Fig. 1 are very different by the look, while other images of *Mr. Bean* and *Sean Bean* are fairly similar as they both wear suits. On the other hand, high level visual object recognition techniques [17,15] focus on detecting objects like bottle, dog, grass, etc. in an image, but are not powerful enough to distinguish entities.

*Context-based methods* use only textual information in the context of the image. Here context refers to URL, descriptive tags for the image, the surrounding text and even search result snippets [14]. To represent the context, all previous work uses bag-of-words or n-grams model [14]. The bag-of-words (BOW) model can not capture the semantics of the context in an accurate way for three reasons. First, limited length of context provide insufficient signals in words model. Second, terms with one or more words are sometimes better semantic units than single words but they are not handled properly by BOW models. Finally, words can be ambiguous. “Apple” may refer to an IT company or a kind of fruit, but BOW model treats all “apple” terms equally. Similar arguments hold for n-gram models.

*Hybrid approaches* attempt to combine the visual features with textual features. However, semantic gaps between the visual and textual features make it difficult to directly combine them into one uniform similarity measure. Some hybrid algorithms therefore resort to co-clustering on visual and text simultaneously such as MMCP [11]. But such approach is iterative, time consuming and thus not suitable for online applications such as image search.



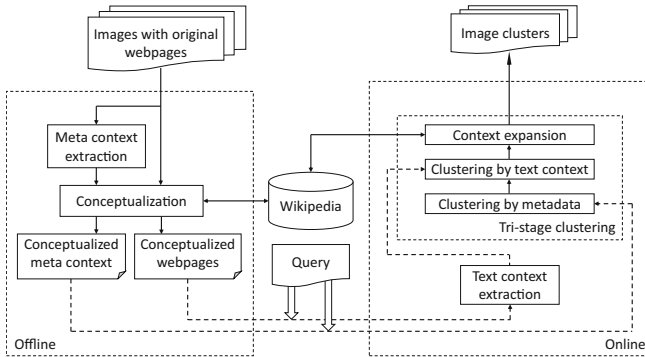
**Fig. 2.** Partial Search Result for “bean” on Prototype System

In this paper, we propose a new context-based approach that emphasizes on understanding textual signals. The reason to focus on text is that, we believe, unlike visual signals, textual signals from the right context explicitly reveal the semantics of the image. Our approach is different from the existing context-based image clustering in three aspects. First, we explicitly disambiguate the context text by converting each phrase to an unambiguous concept from an external knowledge source such as Wikipedia. We call this process “conceptualization”. Conceptualization has been previously shown to be a better way to understand textual signals than bag-of-words model[19]. Second, our method provides concept labels to annotate each cluster of images by accumulating the concepts in the contexts from the clusters. With these labels, users can conveniently grasp what each cluster is about. Third, we propose a modified version of hierarchical agglomerative clustering (HAC) in a tri-stage clustering framework, which is more robust to noise. This framework guarantees the purity of each cluster while improving the inverse purity, i.e. forming as large clusters as possible. The experimental result shows that our approach significantly outperforms competing algorithms, and achieves very high purity, F-measure and NMI scores. A partial result of searching for “bean” on our prototype image search system is shown in Fig. 2. Every cluster shows the most relevant images about a distinct entity, and each cluster is labeled with the 5 concepts which are most related to the entity. The four clusters in Fig. 2 have been correctly identified as *Mr Bean*, *Sean Bean*, *Frances Bean Cobain* and *Phaseolus vulgaris* (the official name for “common bean”).

The rest of the paper is organized as follows. Section 2 presents the structure and each component of our framework; Section 3 demonstrates the experimental results; Section 4 introduces some related work while Section 5 concludes the paper.

## 2 Framework

In this section, we introduce a novel image clustering framework based on conceptualization of contexts. Our input is an image search query and a set of images returned by this query along with their hosting HTML pages. Our output is a number of clusters of images, each containing images of the same entity and each tagged with a concise list of most relevant concepts. For example, the first cluster of Fig. 2 is tagged with “Mr. Bean”, “Rowan Atkinson”, etc.



**Fig. 3.** The Architecture of Image Clustering by Conceptualization

The architecture of our framework is shown in Fig. 3. The framework is divided into two parts: online and offline components. The offline components extract the meta data of the image and conceptualize all of the text in the source page. Online components 1) extract the surrounding text context of the image and query from the conceptualized source page and then use concepts in the context to construct the concept vector representation of the image context; and 2) cluster the images using a tri-stage clustering algorithm. The context extraction process is online because it cannot be done before the query is known. Next, we present each component in more detail.

### 2.1 Context Extraction

This paper concerns two kinds of image context, meta data context and text context. Meta data context extraction is an offline process while text context is extracted online.

*Meta data context* (or meta context in short) are all intrinsic attributes of the image, such as the anchor text of the image (i.e., ALT attribute in image tags) in the web page, the URL of the image. The domain and the file extension in the URLs are ignored because they are less relevant to entity in the image. For example, images from Flickr share the same domain but are not the same entity. We split the URL into “words” by directory separators, special characters or letter case conversion (e.g., from lower to upper case) to get context from URL. In some cases, the URL may contain randomly generated strings:

[http://domain.com/53C316-C2oJ5/AppleInc\\_2012.jpg](http://domain.com/53C316-C2oJ5/AppleInc_2012.jpg)

contains these words: “53C316”, “C2oJ5”, “Apple”, “Inc” and “2012”. Here, “53C316” and “C2oJ5” has no clear meanings, while “Apple”, “Inc” and “2012” are understandable. We extract all 3-grams in each word, such as “C2o”, “2oJ” and “oJ5” in “C2oJ5”, and “App”, “ppl” and “ple” in “Apple”. Each 3-gram corresponds to one feature of this word. Then we learn an L2-SVM model using LIBLINEAR [8] to classify these words and filter out meaningless ones with an accuracy of 95.69%. Note that, using a lexicon such as Wikipedia only does not work because simple strings like “5” or “J” are also valid terms.

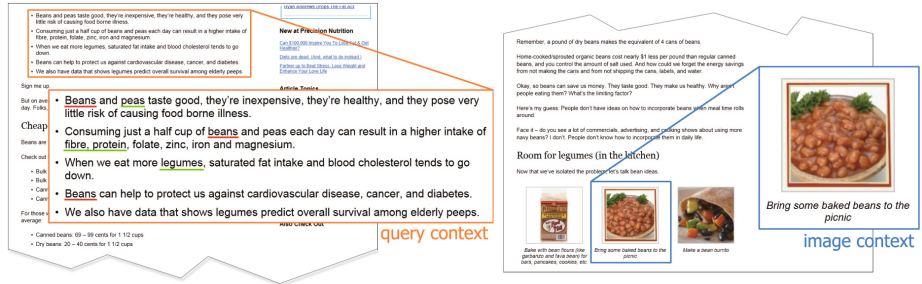


Fig. 4. Image Context and Query Context

Text context is the surrounding plain text of both the image and the query terms in the web page. The reason we employ query context in addition is that the context surrounding the image is likely to be an accurate description of that image but not always enough to distinguish different entities. As Fig. 4 shows, the image context contains limited amount of information. A great deal of signals for identifying “bean” such as “pea (a kind of bean)”, “legume (the family that bean belongs to)”, “fibre (major ingredient of bean)” and “protein (major ingredient of bean)” can otherwise be found in the query context part. We extract the relevant context by a sibling based method [1]. It retrieves all text nodes which contain the query terms, as well as their sibling nodes in the Document Object Model (DOM) tree of the page.

## 2.2 Conceptualization of Context

Wikipedia is a rich and comprehensive knowledge source of concepts. Each concept (e.g. *Mr. Bean* or *Phaseolus vulgaris*) has a descriptive article. The goal of conceptualization based on Wikipedia is to convert a piece of plain text into a set of Wikipedia concepts. To achieve this, we need to recognize the multi-word expressions (MWEs)<sup>1</sup> in the text and then disambiguate them by linking each of them to a corresponding Wikipedia article/concept. Fig. 5 shows an example of conceptualization, where “Polar Bear” is recognized as an MWE and correctly linked to the “Snow Patrol”<sup>2</sup> article.

In this paper, we adopt a conceptualization approach known as *wikification* [5] which is based on link co-occurrence in Wikipedia corpus. The technique first constructs a

<sup>1</sup> MWE is any term that contains one or more words.

<sup>2</sup> Snow Patrol is a Scottish rock band.

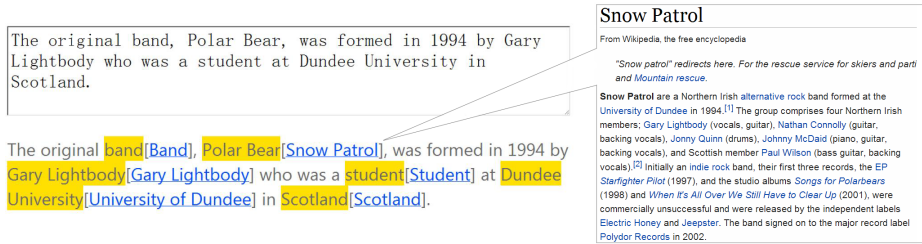


Fig. 5. An Example of Wikification

link co-occurrence matrix iteratively, and then uses the matrix to simultaneously disambiguate all MWEs in the input text by choosing the concept combination that maximizes the likelihood of concept co-occurrence within a sliding window.

### 2.3 Image Clustering

We first introduce the context representation and a modified hierarchical clustering algorithm. We then propose a tri-stage clustering framework.

**Context Representation.** With concepts extracted from the context, we can draw a concept histogram for each image, which represents the image’s semantic information. We use the *vector space model* (VSM) to represent the context. We define a CF-IDF score for each dimension in the concept vector of a textual context. The CF-IDF score of the concept  $c$  in context  $d$ ’s concept vector is adapted from the well-known TF-IDF score in information retrieval, and is defined as:

$$\text{CF-IDF}(c, d) = \text{CF}(c, d) \times \log \frac{|D|}{\text{DF}(c)}, \quad (1)$$

where  $\text{CF}(c, d)$  is the concept frequency of  $c$  in  $d$ ,  $|D|$  is the total number of Wikipedia articles from which we compute the document frequency of each concept while  $\text{DF}(c)$  is document frequency of  $c$ . We compute the document frequency of  $c$  by counting the number of documents which have links to  $c$ .

**HAC with Cluster Conceptualization.** We apply *cosine similarity* to compute the pairwise similarity of contexts. We use a modified HAC algorithm to cluster the contexts. There are two reasons for using HAC: First, we don’t know the exact number of clusters in advance, but we can specify a threshold for minimal similarity within a cluster. Second, HAC is an agglomerative algorithm that merges similar clusters incrementally. Therefore we are able to extend the algorithm by incorporating different features at any step of the clustering process.

There are four common ways to compute similarity between two clusters in HAC: *Single-link*, *Complete-link*, *Group Average*, *Centroid*. These methods compare the individual data points in each cluster without considering each cluster as a whole. This paper adopts a new method to compute cluster similarity. It summarizes the semantic

information in each cluster by building a concept histogram for each cluster. Specifically, given a cluster  $C$  with  $n$  image contexts,  $d_1 \dots d_n$ , the weight of concept  $c$  in the concept vector for  $C$  is

$$V(C)\{c\} = \sum_{d \in C} \text{CF-IDF}(c, d) \quad (2)$$

To restrict the size of this concept vector and to avoid noise, we keep only top  $K$  concepts with the highest weights. The selected concepts and their weights thus represent the semantics of the cluster. This process is called *cluster conceptualization*. The complete HAC with cluster conceptualization (HAC\_CC) is shown in Algorithm 1.  $D$  is the set of images,  $\Pi$  is the set of resulting clusters,  $N$  is the number of images,  $C_i$  is an image cluster,  $V(C)$  is the concept vector of a cluster  $C$ ,  $Sim$  is the function computing the cosine similarity of the two vectors,  $S$  is the similarity matrix of images, and  $\tau_t$  is the threshold that controls the clustering granularity. Line 9 to 15 merge two most similar clusters each time.

---

**Algorithm 1.** HAC with Cluster Conceptualization (HAC\_CC)
 

---

<p><b>Input:</b> Set of images <math>D</math>  <b>Output:</b> Image cluster <math>\Pi</math></p> <pre> 1: function HAC_CC(<math>D</math>) 2:   <math>\Pi \leftarrow \{C_i = \{d_i\} \mid d_i \in D\}</math> 3:   for <math>i \leftarrow 1</math> to <math>N</math> do 4:     for <math>j \leftarrow i + 1</math> to <math>N</math> do 5:       <math>S[i, j] \leftarrow Sim(V(C_i), V(C_j))</math> 6:     end for 7:   end for 8:   for <math>iter \leftarrow 1</math> to <math>N - 1</math> do 9:     <math>max\_sim = \max_{i &lt; j} S[C_i, C_j]</math> 10:    if <math>max\_sim &lt; \tau_t</math> then 11:      return <math>\Pi</math> 12:    end if 13:    <math>C_i, C_j \leftarrow argmax_{C_i \neq C_j} S[C_i, C_j]</math> 14:    <math>C_i \leftarrow Combine(C_i, C_j, S)</math> </pre>	<pre> 15:    <math>C_j \leftarrow \emptyset</math> 16:  end for 17:  return <math>\Pi</math> 18: end function  19: function COMBINE(<math>C_i, C_j, S</math>) 20:   <math>V \leftarrow V(C_i) + V(C_j)</math> 21:   <math>V(C_i) \leftarrow top\ K\ concepts\ of\ V</math> 22:   for <math>m \leftarrow 1</math> to <math>N</math> do 23:     if <math>m &gt; i</math> and <math>m \neq j</math> then 24:       <math>S[i, m] \leftarrow Sim(V(C_i), V(C_m))</math> 25:     else if <math>m &lt; i</math> and <math>m \neq j</math> then 26:       <math>S[m, i] \leftarrow Sim(V(C_i), V(C_m))</math> 27:     end if 28:   end for 29:   return <math>C_i \cup C_j</math> 30: end function </pre>
---	--

---

The advantage of this method is, we can boost the important signals while ignoring noisy ones. On the other hand, since we just keep  $K$  concepts, both cluster similarity and the generation of cluster histogram can be computed in constant time, while HAC using *Group Average* or *Centroid* has a quadratic time complexity to the cluster size.

Similar to the original HAC algorithm, Algorithm 1 has a time complexity of  $O(N^3)$ <sup>3</sup>. We can further optimize it to  $O(N^2 \log N)$  by using a sorted priority queue to store the rows of the semantic matrix  $S$  in line 5, With this optimization, the operation of finding two most similar clusters (line 9) is reduced from  $N^2$  to constant time, and the overall complexity only depends on the sorting process which costs  $O(N^2 \log N)$ .

---

<sup>3</sup> Strictly speaking, it is  $O(K^2 N^3)$ , but  $K \ll N$  so it is treated as a constant.

**Tri-stage Clustering.** Generally speaking, meta context is the most reliable image context since it is guaranteed to be related to the image, whereas the text context may contain noise. As such, we use these two kinds of context at different stages of clustering. Further, to remedy insufficient signals, we expand the contexts by using additional information from Wikipedia, and perform the third stage of clustering. The above stages form a tri-stage clustering algorithm which includes *meta context clustering*, *text context clustering* and *expansion clustering*.

In the first stage, we construct the concept vector of each image using the concepts extracted from the URL and anchor texts, and apply the HAC\_CC algorithm on the images. Although the signals from meta data are reliable, useful signals are limited. Thus, many small clusters are formed with very high purity.

In the second stage, we merge the concept vector extracted from the text context into the concept vector of meta context for each image and combine all the vectors for each cluster from stage one to obtain the cluster vectors (Eq. (2)). We again apply HAC\_CC algorithm on these new cluster vectors. Only top 50 concepts in each resulting cluster are kept to filter out the noise.

The final stage takes as input the clusters formed in the second stage, and expands the context of each cluster in an attempt to merge some of the clusters which should have been together. For each of the top  $K$  concepts in a cluster, we extract the top 50 concepts (ranked by CF-IDF) from the Wikipedia article of that concept, and replace the concepts in the previous stage with them. The weight of the concept  $c$  in the new vector  $V'(C)$  is defined as:

$$V'(C)\{c\} = \sum_{c_i \in V_C} (V(C)\{c_i\} \times \text{CF-IDF}(c, d_{c_i})), \quad (3)$$

where  $V_C$  is the previous concept vector of cluster  $C$ ,  $c_i$  is one of the concept in  $V_C$ , and  $d_{c_i}$  is the Wikipedia article of  $c_i$ . After reconstructing the new concept vector, HAC\_CC is again applied to form the final clusters.

When the third stage finishes, we rank the concepts (dimensions) in the aggregated concept vector of each cluster by the values and use top concepts to represent the semantics of that image cluster. The complexity of the tri-stage clustering algorithm remains the same as HAC\_CC algorithm because the input size of each stage is bounded by the total number of images.

## 2.4 Use Scenario

Our framework has an online component because the query terms, which are important signals for context extraction, must be processed at runtime. Although the clustering algorithm presented earlier has a non-linear time complexity, the following use case of our framework is typical and practical. User enters a search term and the search engine returns a number of relevant images on page-by-page display. On any given page, the user can choose to “order by entity”, and the clustering framework will re-organize the results on that page (typically a few tens to several hundred images) by entities, as shown in Fig. 2. This is practical because, as we will show later, the online part of the algorithm completes within a second for 100 images.



### 3 Experimental Results

This section evaluates the image clustering system. We first present the experiment set-up and evaluation metrics. Then, we show four experiments. The first experiment evaluates the performance of each key component of our system. The second one gives an end-to-end comparison between our approach and the state-of-the-art systems. The third one illustrates the accuracy of concepts generated by our system for each cluster. The last one evaluates the time efficiency of the system.

#### 3.1 Experiment Setup

We prepare an image data set from Google Image Search, sorted by relevance. We select a list of 50 ambiguous queries as shown in Fig. 6 (10 for parameter training and 40 for testing). For each query, we query in Google Image and download the top 100 images returned by Google with the original web pages of the images. This data set contains a total of 5,000 web pages/images. We then ask two human judges to manually cluster the collected data to create two label sets. All evaluation metrics computed in subsequent experiments are the averaging values over these two sets. All experiments were run on a dual-core Intel i5 machine with 14GB memory.

barcelona, berry, curve, david walker, diff, george foster, john smith, longhorn, manchester, puma
acrobat, adam, amazon, anderson, andrew appel, apple, arthur morgan, bean, british india, carrier, champion, eclipse, emirates, explorer, focus, friends, jaguar, jerry hobbs, jobs, kiwi, lotus, malibu, morgan, nut, palm, patriot, perfume, pluto, polo, santa fe, shell, sigma, studio one, subway, taurus, tick, tucson, venus, visa, wilson

Fig. 6. Queries for training(above) and testing(below)

#### 3.2 Evaluation Metrics

We adopt three well-known metrics to measure the result of image/document clustering: *Purity*, *NMI* and  $F_1$ . *Purity* measures the intra-cluster accuracy. It has an obvious drawback that if we create one cluster for each document, the *Purity* will be 1, and this is not useful at all. Therefore, *Purity* should not be viewed independently. *NMI* (Normalized mutual information) is a better measure that balances the purity of the clusters with the number of clusters. It measures the amount of common information between the computed clusters and the ground truth. Another measure of clustering is  $F_1$  score, which combines *Purity* and *Inverse Purity*. *Inverse purity* exchanges the position of the result and the ground truth in the the purity computation, and determines how much of each cluster in the ground truth is correctly clustered together. Similar to the  $F_1$  score used in information retrieval task,  $F_1$  score is computed as:

$$F_1(C, L) = \frac{2 \cdot Purity(C, L) \cdot Purity(L, C)}{Purity(C, L) + Purity(L, C)}, \quad (4)$$

where  $C$  is the clustering result and  $L$  is the ground truth clusters. In many studies of clustering algorithms, *NMI* is more important and sometimes the only measure, because it's extremely difficult to achieve high *NMI* scores.

### 3.3 Threshold of Tri-stage Clustering

The tri-stage clustering (TSC) algorithm is based on HAC\_CC algorithm. Similar to traditional HAC algorithm, HAC\_CC has a threshold to control the granularity of the clustering result. We tune different threshold  $\tau_t$  of HAC\_CC on a training data collected from top 100 images of 10 different queries. Cluster labels are assigned to each image by human judges. Fig. 7 shows the clustering result on different thresholds of HAC\_CC. We prefer to choose a threshold which can ensure high purity, F1 and NMI at the same time. NMI reaches a peak value at  $\tau_t = 0.15$ . At this threshold, the purity is significantly higher than when  $\tau_t = 0.1$  and F1 score is relatively high, too. Consequently, in this system, we set  $\tau_t$  to be 0.15.

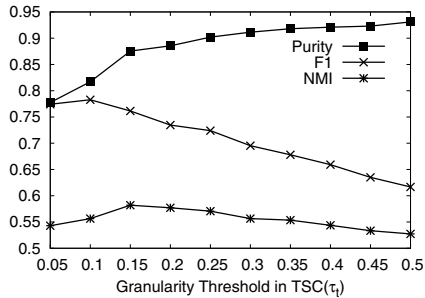


Fig. 7. Clustering Result on Different  $\tau_t$

### 3.4 Evaluation on Key Components

In this sub-section, we experiment on different variants of our system. First, we investigate the effects of different context extraction methods. Then we show the performance of concept representation based on conceptualization. Finally, we show the benefits of tri-stage hierarchical clustering.

**Context Extraction:** There are three variants of context: the whole page (Page), surrounding text of the image (Image) and surrounding text of both the image and query terms (I & Q). The window size of the surrounding text is empirically set to 200 words (100 words before and after the query/image respectively). Table 1(a) compares the end-to-end results of image clustering on 20 different queries using these three types of context. One can stipulate that the noise in whole page contexts adversely affect the purity of the clusters. Even though the surrounding text of the images already gives rise to very pure clusters, adding the query context gives better F1 and NMI. Overall, the text context of both image and query terms wins because of superior cluster accuracy at limited computation overhead.

**Context Representation:** We implement two baseline systems to compare with our concept vector (CV) model. One of them uses bag-of-words(BOW) model and the other one uses bag-of-phrase(BOP) model. The latter is a minor enhancement to BOW, and uses (possibly ambiguous) MWEs instead of single words to represent the context. Different from these two baselines, our system disambiguates MWEs in the context to

**Table 1.** Comparison on Key Components

(a) Diff. Contexts				(b) Diff. Representations				(c) Diff. Algorithms				
	Purity	F1	NMI		Purity	F1	NMI		Purity	F1	NMI	Time
Page	0.71	0.78	0.35	BOW	0.92	0.54	0.48	AP	0.92	0.55	0.50	1.9s
Image	<b>0.91</b>	0.80	0.59	BOP	<b>0.94</b>	<b>0.62</b>	0.50	HAC	<b>0.94</b>	0.62	0.55	0.9s
<b>I &amp; Q</b>	0.90	<b>0.81</b>	<b>0.62</b>	CV	<b>0.94</b>	<b>0.62</b>	<b>0.55</b>	HAC_CC	<b>0.94</b>	0.76	0.59	0.7s
								TSC	0.90	<b>0.81</b>	<b>0.62</b>	1.1s

generate a more accurate representation. To make the end-to-end results comparable, we apply HAC on all three types of representations, since the tri-stage clustering algorithm is only applicable to our CV model. Table 1(b) shows the comprehensive clustering results. This experiment shows that the BOP/CV representations are much more effective than BOW, with particular improvement in F1 score. Phrases are more accurate to identify the semantics of text than single words. CV beats BOP on NMI because it disambiguates the MWEs in the context and thus makes the similarity computation between two images more accurate.

**Tri-stage Clustering:** We compare HAC\_CC and TSC with HAC and Affinity propagation (AP), two very popular clustering algorithms. In this experiment, all algorithms use the concept vector representation. Except for TSC which clusters in three stages, all other algorithms run one time only. The threshold  $\tau_t$  of HAC and HAC\_CC is set to 0.15, while the preference of AP is set to the average similarity between the data points. We also report the time cost of the algorithms by averaging 5 independent runs in the same setting. The result of these three algorithms are shown in Table 1(c). HAC\_CC algorithm outperforms AP and HAC due to the enhancement of strong signals and removal of noise in cluster conceptualization process. TSC further improves HAC\_CC with concept expansion because 1) we make use of meta context, and 2) the previous clustering stage provides accurate cluster vectors as input to the next stage to further reduce the influence of noise. The experiment demonstrates TSC’s capability of boosting important semantic signals which substantially helps improve the accuracy of web image clustering.

### 3.5 End-to-end Accuracy

We compare our approach (TSC) with two image clustering systems and two text clustering systems from the literature (See Table 2). The first image clustering system is implemented following Cai’s [3] approach, which extracts image context using VIPS [4]. The second image clustering system is the multi-modal constraint propagation approach (MMCP) [11]. We also compare with text clustering systems as baselines because our approach only extracts text features from the image context and therefore can be considered as text clustering as well. The two text-based methods that we compare with are HAC clustering on bag-of-words (BOW) and HAC clustering of topics extracted by LDA[2], and both are input with the same text context used in our algorithm, i.e., meta context and text context concatenated in one blob.

Cai’s system used visual features, textual features (context), and an image link graph. They used Color Texture Moments[25] as visual features and bag-of-words in the visual context as textual features. We replicate the link graph from a subset of source pages without obtaining the entire set of web pages, according to the property described by Cai. For MMCP, we apply the same modalities mentioned by Fu: local visual, global visual and text. Fu used tags of the images in Flickr as the textual features. However, without available tags, we instead use the bag-of-words in the source page of the image.

The two text clustering systems use different representations for the text context (i.e., BOW and topics) to compute the similarity between two image contexts, and then use HAC algorithm to cluster the contexts. In the LDA system, we directly extract topics in the test data. The parameters of each system are tuned to the one that maximizes the NMI score in the training data. The clustering threshold  $\tau_t$  is set to 0.2 in BOW baseline and 0.25 in LDA. The number of topics for LDA is set to 150.

The four competing systems generally do not have a good way of handling noise, which is often seen in the contexts of web images. The noise usually dilutes the positive impact of the important signals, especially when the context is of limited size. Our conceptualization and tri-stage clustering method can help remove some of the noise. Some systems like MMCP intends to obtain high NMI score, but their purity is very low. The BOW system achieves the highest purity because of the exact match of the words in the context, but otherwise has a low F1 score. In contrast, the LDA system has some degree of generalization which makes it perform better than BOW in F1 scores. However, LDA failed to capture high quality topics for images that have very short and noisy contexts. Consequently, it has relatively poor purity. Over all, our approach outperforms other systems by producing bigger clusters while preserving the high purity in each clusters. It defeats the best of the peers by significant margins: **17.4%** by  $F_1$  and **29.2%** by NMI score.

**Table 2.** Results of End-to-End Image Clustering

	Purity	F1	NMI
Cai	0.60	0.71	0.10
MMCP	0.74	0.58	0.34
BOW+HAC	<b>0.92</b>	0.54	0.48
LDA+HAC	0.88	0.60	0.44
<b>TSC</b>	0.90	<b>0.81</b>	<b>0.62</b>

### 3.6 Cluster Conceptualization Accuracy







In this subsection, we show the conceptualization result on the test queries. To quantify the accuracy of conceptualization on all 40 test queries, we manually label the results in the following manner. For the top 5 clusters of each query, we pick top ten ranked concepts for each cluster and judge whether the concept is relevant to the images in the cluster by human. This results in around 2000 concepts to be labeled. Each query is labeled by three persons and the accuracy for each image clusters is averaged on the judgement from the three persons. Formally, the accuracy of conceptualization of an image cluster is defined in Eq. (5).

$$Accuracy(C) = \frac{1}{M} \sum_{i=1}^M \frac{1}{|C|} * \sum_{c \in C} f_i(c), \tag{5}$$

where  $C$  is the set of concepts for an image cluster,  $M$  is the number of the human judges ( $M = 3$  in our experiment), and  $f_i$  is the judgement of the  $i^{th}$  judge. If concept  $c$  is labeled as relevant to the cluster,  $f_i(c) = 1$ , otherwise  $f_i(c) = 0$ . We average the accuracy of all clusters on the test queries, and the final result is **71.82%**.

Table 3 shows some examples of our conceptualization results. For each query, we show only the first two clusters as well as the most related concepts generated from different entities. Terms listed under the images are 5 top-ranking Wikipedia concepts that are conceptualized from each image cluster. Each of the concept has a corresponding Wikipedia article. For example, the concept ‘‘Kiwi’’ in Wikipedia is the bird kiwi, while ‘‘Kiwifruit’’ refers to the fruit kiwi.

**Table 3.** Conceptualization of Image Clusters (Adam, Eclipse, Kiwi)

Query	Cluster
Adam	
	Adam Lambert, American Idol, God, Kris Allen, Privacy policy
Adam	
	Adam Levine, Hijab, Mehndi, Fashion, Hairstyle
Eclipse	
	Solar eclipse, Sun, Moon, Lunar, Umbra
Eclipse	
	Twilight (series), Bella, David Slade, Vampire, Stephenie Meyer
Kiwi	
	Kiwifruit, Fruit, Recipe, Health benefit, New Zealand
Kiwi	
	Kiwi, Bird, New Zealand, Egg, Smithsonian National Zoological Park

### 3.7 Time Efficiency

First, We evaluate the time cost of the online and offline components in our system. The results are averaged over 5 independent runs, on the 40 test queries. The average execution time per query (with 100 images to cluster) of offline and online components are 471 seconds and 1 second, respectively. The off-line component consists of image

context extraction, chunking, and conceptualization, of which conceptualization is the most expensive process. The current offline-online split of the system effectively pushes the most time consuming work to the preprocessing stage and thus makes the online part more efficient and practical.

Second, we compare the average online clustering time of our system (1121 ms) with MMCP (5021 ms) and Cai's system (194 ms). All timing results are averaged over 5 independent runs. MMCP propagates the constraints among modalities. This process clusters on each modality for several times, which explains its long execution time (5 seconds). With all features extracted off-line, Cai's system only need spectral clustering on the images online, which explains why it is the winner here. However, the VIPS extraction module of Cai's system relies on the browser rendering module and crashes frequently. It is almost impossible to automate the context extraction process without human intervention. Our prototype system, which is not optimized, runs for around 1 second per query on average. It is slower than Cai's since we need to extract the query context online, and the expansion of concepts is also time consuming. However, with accuracy, efficiency and reliability all considered, our system is an overall winner in practical web image search tasks.

## 4 Related Work

We divide existing image clustering methods into three categories: content-based, context-based and the combined approaches.

Content-based image clustering approaches [10,7,13] rely on visual signals. For example, Fu et al.[11] gave a constraint propagation framework for multi-model situations. They constructed multiple graphs, one for each visual modalities such as color histogram, SIFT descriptors [18], etc. The nodes are images while the edges are similarities between the images by a particular visual modality. A random walk process is employed on these graphs. All of the above work uses low-level visual signals of images such colors, gray scales, contrasts, patterns, etc. These signals are insufficient to capture high level semantics of the images. This is evident from our experiments on Fu's algorithm which heavily relies on basic visual signals. There has been some development on high level visual object recognition and semantic annotation [17], but even the state-of-the-art techniques in this area suffer from low accuracy and unreliability.

With the difficulty in content-based clustering, some researchers turn to signals coming from the context of the images, such as file name, alternate text and surrounding text. Cai et al. made some progress in this respect. They represented a web page segmentation algorithm named VIPS [4], which works by rendering the web page visually and detecting the important visual blocks in the page. And they subsequently proposed three kinds of representations for images [3]: visual feature based representation, textual feature based representation and link graph based representation, and proposed a two-level clustering algorithm which combined the latter two. Jing et al. [14] introduced a novel method named IGroup for image clustering. Instead of clustering on returned images directly, they first search the query on normal web search engine, and cluster the titles and snippets from the search results. They then construct a new query string to represent each of the cluster, and send these query strings to the image search engine to get images for each cluster. To construct the query string, they used an algorithm proposed by

Zeng[26]. These bag-of-words approaches are inadequate for understanding the semantics of the context. Relying on bag-of-words or n-grams can easily confuse noise with meaningful signals. Our approach, on the other hand, leverages co-occurrence information on high level concepts mined from Wikipedia, a comprehensive knowledge source, and most importantly, is able to disambiguate entities using this knowledge. Hence, we are able to achieve better results.

Recently there are many attempts on combining visual features and textual features in image clustering. Feng et al.[9] used the surrounding text of images and a visual-based classifier to build a co-training framework. Gao et al.[12] represented the relationship among low-level visual features, images and the surrounding texts in a tripartite graph. Wang et al.[24] reinforced visual and textual features via inter-type links and inversely uses those features to update these links. The visual features, text features and inter-type links are represented as three matrices. Three linear formulas is defined to iteratively update the three matrices. Ding et al.[6] proposed a hierarchical clustering framework. Leuken et al.[16] investigated three methods for visual diversification of image search results in their paper. Tsai et al.[23] proposed a technique based on visual synset for web image annotation. They applied affinity propagation clustering on a set of images associated with a query term based on both visual and textual features. Each cluster represents a visual synset, and is labeled by related query terms. However, this query-based/term-based labeling approach has two limitations: 1) it cannot produce related concepts to the clusters like our system does (e.g. “Teddy” for Cluster 1 in Fig. 2); 2) the related query terms themselves can be ambiguous and are not suitable for representing a visual synset. In our paper, we represent each cluster with high related concepts which are Wikipedia concepts without ambiguity. The main challenge with the above hybrid approaches is the semantic gap between visual signals and textual signals. There is no easy way to combine the two kinds of similarity measures into one unifying measure.

## 5 Conclusion

In this paper, we proposed a novel framework for clustering web images by their contexts. The novelty lies in that our framework seeks to “understand” a context by converting words and phrases in the context into high level concepts in an external knowledge base such as Wikipedia. Moreover, it performs a tri-stage modified HAC algorithm utilizing information of various reliability. Our experiments show that on 40 “ambiguous” query terms, the purity, F-measure and NMI of our clustering results are consistently better than other recently developed image clustering systems. Our prototype system is practical as it is able to cluster a page of 100 images within 1 second.

## References

1. Alciç, S., Conrad, S.: Measuring performance of web image context extraction. In: MDMKDD, vol. 8, p. 8 (2010)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022 (2003)

3. Cai, D., He, X., Ma, W.Y., Wen, J.R., Zhang, H.: Organizing www images based on the analysis of page layout and web link structure. In: ICME, pp. 113–116 (2004)
4. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: VIPS: a vision-based page segmentation algorithm. In: Microsoft Technical Report, MSR-TR-2003-79 (2003)
5. Cai, Z., Zhao, K., Zhu, K.Q., Wang, H.: Wikification via link co-occurrence. In: CIKM, CIKM 2013, pp. 1087–1096 (2013)
6. Ding, H., Liu, J., Lu, H.: Hierarchical clustering-based navigation of image search results. In: MM, pp. 741–744 (2008)
7. Fan, J., Gao, Y., Luo, H.: Hierarchical classification for automatic image annotation. In: SIGIR, pp. 111–118 (2007)
8. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9, 1871–1874 (2008)
9. Feng, H., Shi, R., Chua, T.S.: A bootstrapping framework for annotating and retrieving www images. In: MM, pp. 960–967 (2004)
10. Fergus, R., Li, F.F., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: ICCV, pp. 1816–1823 (2005)
11. Fu, Z., Ip, H.H.S., Lu, H., Lu, Z.: Multi-modal constraint propagation for heterogeneous image clustering. In: MM, pp. 143–152 (2011)
12. Gao, B., Liu, T.Y., Qin, T., Zheng, X., Cheng, Q., Ma, W.Y.: Web image clustering by consistent utilization of visual features and surrounding texts. In: MM, pp. 112–121 (2005)
13. Gao, Y., Fan, J., Luo, H., Satoh, S.: A novel approach for filtering junk images from google search results. In: MMM, pp. 1–12 (2008)
14. Jing, F., Wang, C., Yao, Y., Deng, K., Zhang, L., Ma, W.Y.: IGroup: web image search results clustering. In: MM, pp. 377–384 (2006)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
16. van Leuken, R.H., Pueyo, L.G., Olivares, X., van Zwol, R.: Visual diversification of image search results. In: WWW, pp. 341–350 (2009)
17. Li, L.J., Socher, R., Li, F.F.: Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: CVPR, pp. 2036–2043 (2009)
18. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
19. Song, Y., Wang, H., Wang, Z., Li, H., Chen, W.: Short text conceptualization using a probabilistic knowledgebase. In: IJCAI (2011)
20. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW, pp. 697–706 (2007)
21. Taneva, B., Kacimi, M., Weikum, G.: Gathering and ranking photos of named entities with high precision, high recall, and diversity. In: WSDM, pp. 431–440 (2010)
22. Taneva, B., Kacimi, M., Weikum, G.: Finding images of difficult entities in the long tail. In: CIKM, CIKM 2011, pp. 189–194 (2011)
23. Tsai, D., Jing, Y., Liu, Y., Rowley, H., Ioffe, S., Rehg, J.: Large-scale image annotation using visual synset. In: ICCV, pp. 611–618 (2011)
24. Wang, X.J., Ma, W.Y., Zhang, L., Li, X.: Iteratively clustering web images based on link and attribute reinforcements. In: MM, pp. 122–131 (2005)
25. Yu, H., Li, M., Zhang, H.J., Feng, J.: Color texture moments for content-based image retrieval. In: International Conference on Image Processing, pp. 24–28 (2003)
26. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: Learning to cluster web search results. In: SIGIR, pp. 210–217 (2004)
27. Zhong, S., Liu, Y., Liu, Y.: Bilinear deep learning for image classification. In: MM, pp. 343–352 (2011)