Good afternoon ladies and gentlemen, welcome and thank you for your interest in our paper. My name is Kaiqi Zhao. I was a student from Shanghai Jiao Tong University and now I am a Ph.D student at Nanyang Technological University. I am very pleased to present our research here. My title is “Clustering Image Search Results by Entity Disambiguation”.

(Slide 2)

Here’s the outline of my presentation. At the beginning, I will introduce our motivations and existing research on image clustering. After that, I am going to present our framework and then show some experimental results.

(Slide 3)

OK, let’s start with an example to illustrate the motivation of our research problem. John, the guy sitting at the computer, is searching for images of beans to prepare his slides for a biology course. He inputs the query term “bean” and Google returns him the results as a mixture of several entities related to bean, including 5 related persons and the bean he wants to search. He feels difficult to look for images of beans, to say nothing of choosing proper images from them.

(Slide 4)

Consider another kind of searching result presentation. If the image result is clustered, John may feel convenient to find out images for beans and know what the other entities are by checking the representative terms that used to characterize the entities. John can even look for more images about one entity if he click on one of the cluster, then the system uses those representative terms as keywords to search for images that highly related to the entity that John clicks.

Therefore, in our research, we aim to provide two outcomes in our framework. One is the clusters of entities, the other is the representative terms for each entity.

(Slide 6)

Now, let’s turn to some existing image clustering techniques. We divide them into 3 categories. The first one is content-based approaches. These approaches make use of the visual signals of the images, including local features such as SIFT, edge histogram, etc., and global features such as color, contrast, gray scale, brightness, and so on. The second type of approaches are context-based approaches, which usually use the bag-of-words representation of the context of the images. The context can be the surrounding words as well as the visual blocks that the image belongs to in the source web page. The third type of approaches are hybrid approaches, which use both text features and visual features, for example the recently proposed multi-modal constraints propagation method.

(Slide 7)

However, the existing content-based approaches rely only on visual signals which may not be reliable in some scenarios. For example, the shape features like SIFT or edge histogram cannot distinguish the two figures in the red boxes. Also, several images of “Mr. Bean” are very different by the look, the visual signals may not provide enough hints to group them as one entity. The existing context-based methods use bag-of-words representation and they do not disambiguate the terms in the context. This may limit the clustering accuracy. The hybrid methods suffers from the semantic gap between the visual and textual signals, so that we cannot combine them into a uniform similarity measure. Some co-clustering methods try to overcome the semantic gap but have high time consumption.

In our research, we focus on a framework which makes full use of the context for clustering images to different entities and provide explanation of the meaning of each entity.

(Slide 8)

There are three challenges in solving our problem, the first one is to define the context of an image. The second one is how we organize and use the context, and the last is how to generate the representative terms.

We propose a framework the overcome these challenges.

(Slide 10)

We use this graph to illustrate the structure of our system. It comprises two parts, online and offline. The offline part is used to extract the meta information of the images and convert all the texts in the source pages to concept space using a process called conceptualization. When the query comes, the online part extracts the conceptualized context of the image, and then passes to the tri-stage clustering algorithm. Notice that, the meta data can be extracted offline while the text context cannot be, because it includes the surrounding text for the query term in the source page. I will introduce the text context in the next slides.

(Slide 11)

We use two types of context, the meta context, and the text context. The meta context contains URL, alternative texts or title text of an image in the html source. URLs often contain random strings which have no exact meaning. We extract terms by splitting the URL using separators and detect valid terms using a binary classifier trained on tri-grams.

(Slide 12)

The text context contains image context and query context. As shown in the following figure, the image context are the texts surrounds the image while the query context are the texts surrounds the query term.

(Slide 13)

In order to understand the context, we want to identify each term extracted to an entity in the knowledge base, so that each term has exactly one meaning and we can use the knowledge base to expand the context to solve the problem of insufficient context. We link the terms in the text to Wikipedia concepts, using an entity-linking technique, and we call this process conceptualization. For each term, there are several candidate Wiki concepts. By considering the context, we can select the correct one, and link the term to a Wikipedia article, which describes a concept. For example, here, we link the term feed to the article titled “forage”.

(Slide 14)

With the Wiki concepts, we use a TF-IDF like scheme to represent the context of the images. The score, called CF-IDF here, comprises the concept frequency in the context of the image, denoted by “CF”, as well as the inverse context frequency for the concept, denoted by “DF”.

(Slide 15)

With the concept representation, we can compute similarity, and use clustering algorithms like k-means, or hierarchical clustering algorithms. Here, we consider hierarchical agglomerative algorithm, HAC. HAC repeatly combines the most similar two clusters at each time, and stop when the largest pairwise similarity is smaller than a predefined similarity threshold.

(Slide 16)

To compute the similarity between clusters, there are four frequently used methods. However, these methods do not consider the cluster as a whole, and are not able to generate representative terms for each cluster. Therefore, we use cluster conceptualization to obtain a concept vector for each cluster. Basically, we combine all the concept vectors of the images in the cluster and keep the strongest ones because the rest ones are not essential to the cluster and may be noise concepts.

We call the HAC using cluster conceptualization as HAC\_CC for simplicity in the following slides.

(Slide 17)

Now, we turn to the tri-stage clustering framework. The idea is, we start from the meta context, which is the most related feature to an image, because some meta contexts may contain the name or very high related terms to the entity. We first cluster using the meta context, then based on the result, we cluster using the text context, and then based on the result, we further expand concept vector using Wikipedia, and combine similar clusters.

When we expand the concept vector after some clustering results, the similarity would become larger because we have more common concepts. Then we can resume the HAC algorithm to combine some similar clusters after the expansion. Notice that, it is different to directly use all the features to build a large concept vector. We keep only the strongest concepts and remove noise concepts at each stage to ensure high accuracy.

(Slide 18-20)

Specifically, in the first stage, we build a concept vector using the concept extracted from meta context, and then perform HAC\_CC. In the second stage, we expand the concept vector by the text context, and then resume HAC\_CC. In the third stage, we replace the each concept in the vector by the concepts in its Wikipedia page, and combine using a weighted sum schema. Then we resume HAC\_CC.

(Slide 21)

We show here an example of merging clusters through the three stages. The concepts in red color represents the new concepts after expansions. In the first stage, image 1 and 2 both has concepts “mr. bean” in meta context, we merge them to a cluster. By expanding using text context, “Rowan Atkinson” appears in the vector of the cluster. Then we put image 3 to the cluster because they both have concept “Rowan Atkinson”. In the final stage, we expand the vector by more related Wikipedia concepts, for example add “Blackadder” to the vector, then we now can put image 4 to the cluster. From the result of the tri-stage clustering algorithm, we pick the top-k concepts in the concept vector of each cluster as representative concepts.

(Slide 22-23)

We run the experiments on 50 ambiguous queries listed below, and for each of the query, we retrieve 100 images from Google Image Search. We manually label the cluster of all of the 5000 images. We use purity, F1 and NMI to evaluate our clustering accuracy. Purity measures the number of correct images in the clusters. F1 considers both purity, and inverse purity, which constrains the size of each cluster could not be too small, otherwise the inverse purity would be low. NMI measures the common information between the ground truth and results. Notice that, NMI and F1 are more proper to measure the overall result, while the purity measures intra-cluster accuracy.

(Slide 24)

We show here the effectiveness of each component. Using surrounding text of both image and query is more effective than only use those of image. We use different representations in the original HAC algorithm and found that concept representation is more accurate in NMI compared to bag-of-phrase representation which is an improvement over bag-of-words. The tri-stage algorithm enjoys overall better accuracy because of than other baselines including affinity propagation, HAC and single-stage HAC\_CC on the mixture of all features. This result shows the effectiveness of using TSC to filter noise concepts. The time here is counted for clustering 100 images. TSC is not much slower than HAC and HAC\_CC because they all combine two clusters in one iteration and without backtracks.

(Slide 25)

We also compare our framework to several existing image clustering techniques, and found that TSC has much higher accuracy in clustering web images to different entities than other methods.

(Slide 26)

At last, I show some results of clustering and representative concepts here to demonstrate the quality of our framework. Due to space limitation, I only list two clusters for each query. The representative concepts are the top 5 concepts in the concept vector of each cluster.