Semantic and Visual Image Clustering

Retrieving Search Term Related Pictures in Structured Clusters

Seminar paper

Semantic Multimedia

Summer Term 2013

Hasso-Plattner-Institut für Softwaresystemtechnik GmbH

Universität Potsdam

written by

Mandy Roick
Claudia Exeler
Tino Junge
Nicolas Fricke

30. August 2013

Abstract

Abstract goes here.

Write it at the end.

Write the abstract

Contents

Todo list

1 Retrieving Images in Clusters

introduce the introduction?

1.1 Problem Statement & Motivation

Many semantic image analysis algorithms, e.g. for image categorization or content detection, require training data on which the relevant features can be learned. Obtaining such training data can be a troublesome task, especially when the training set is created manually from scratch. If, for example, an algorithm shall be trained to identify and categorize kinds of food, one would have to think of all possible kinds of food, search for corresponding images and divide them into homogeneous groups.
A good place to search for images are online photo communities like Flickr[1], which provide vast amounts of collaboratively tagged images. These communities are also called folksonomies, i.e. socially indexed collections. Although folksonomies can be good sources for training data due to the semantic metadata that tags provide, several problems exist: First of all, annotations are often of poor quality, since anyone can tag anything without any control mechanisms. Secondly, one can only search for a specific term, and will obtain images for all of the term's meanings, while on the other hand the retrieval is limited to those images which are annotated with the exact same tag. This results to the fact, that usually no semantically relations to other terms are provided. Furthermore, the images will be of a very large visual diversity, which is often not desired.

This work presents a tool whose main aim is to create homogeneous groups of semantically and visually similar pictures for a given topic in order to aid with the laborious assembly of training and test data sets. It is based on the 1 million images of the MIRFLICKR-1M[2] file set, and addresses the above mentioned difficulties. The main challenges encountered were homonymy of keywords (the fact that one word can have multiple meanings), low quality of tags and other annotations, and the consideration of both semantic and visual information about a picture.

1.2 Clustered Tree Nodes Approach

The tool has been implemented as a Python web application, using WordNet[3] for semantic image analysis, SimpleCV[4] for visual image analysis and Flask[5] for the frontend presentation.
It provides ready-to-use semantically and visually homogeneous image clusters for a given topic, or search term. This is achieved by 2 major phases: First, spanning a tree of subordinate terms of the topic and retrieving related images by their keywords for each node of the tree. Second, clustering the images by their predominant keywords as well as by

[1] https://www.flickr.com/
[2] http://press.liacs.nl/mirflickr/
[3] http://wordnet.princeton.edu/
[4] http://www.simplecv.org/
[5] http://flask.pocoo.org/

colors and edge structure. The following figure 1 illustrates these two main phases of the tool.

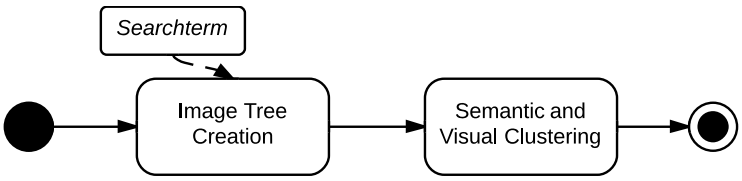

Figure 1: The two main phases of the clustered image search

After giving an overview of Related Work in chapter 2, we present how we analyze the image annotations and the user's search term to retrieve relevant images in chapter 3. The methods applied to cluster the images semantically and visually are described in chapter 4. Chapter 5 explains how we evaluate our approach, while the evaluation results are discussed in chapter 6. At last, chapter 7 gives ideas for improvement and possible future work.

2 Related Work

Much research has been done recently in image clustering and semantic clustering, with application areas in image segmentation, compact representation of large image sets, search space reduction and avoiding the semantic gap in content based image retrieval (LKI11).
However, most of this work focuses on new algorithms for one of the above use cases, not on methods to generate training data.

One algorithm for retrieving training data for image analysis is presented in (OW10). The algorithm collects training data for computational analysis of the quality of photographs from Flickr. But, instead of analyzing pictures automatically according to their tags and comments, the paper presents a tool for collecting user votes. Comments were only used to retrieve terms for describing image quality.

Some of the closely related subjects like Semantic Clustering and Content-Based Image Retrieval are presented in this chapter.

2.1 Semantic Clustering and Tags

The idea of clustering search results based on tags and other annotations has been implemented before by (RHMGM09) but for web pages instead of images. The main difference is that documents such as web pages consist of words, so their content itself can be used for semantic analysis. Current issues with tag-based search and clustering are mostly related to the lack of a defined tag vocabulary (e.g. the use of synonyms, homonyms, variations in spelling etc.), and elaborated on more closely in (RBV$^+$11).

There are already some approaches using WordNet for finding semantic similarities between different Words, like (**?**). They motivate using a knowledge base like WordNet to deal with general problems of the natural language, facing different meanings of a word and strong relations between different words. With the help of a hierarchical concept graph (HCG), consisting of hyponymns and meronyms for a specific entity, the semantic similarity between different words can be determined.
Another approach is using a WordNet-based clustering technique for text documents, which is described in (**?**). They tried to solve the problem of synonyms and ambiguity of words within texts, while adding a part-of-speech tag to every word based on knowledge provided by WordNet. Unfortunately the found out, that including synonyms and hypernyms does not improve the effectiveness of clustering, which they mainly relate to noise which comes with incorrect word interpretation of WordNet.

Conclusion, Problems?

More semantic related word?

2.2 Image Annotation and Content-Based Image Retrieval

Ideas exist to use visual features to semantically analyze and classify images. (LZLM07) and (ZIL12) provide good summaries and evaluations of the different approaches how

this could be done. Both conclude that this so-called *Automatic Image Annotation* is computation-intensive and not yet fully mature.
One approach that combines semantics and visuals to analyze pictures in a so called *visual folksonomy* is (LMS+09). The idea is trying to annotate images, with a controlled vocabulary, based on visual features and existing tags. Their goal, however, is to create additional annotations for not or poorly tagged images.
Another approach is presented by (CHL+04) with the aim to cluster images returned by a WWW image search. In contrast to pictures from a folksonomy, image search results are connected to a web page with context and link information which is used by the algorithm to cluster images.

Include more Related Work for Image Annotation

3 Image Tree Based on WordNet

This chapter describes how the WordNet ontology is used to retrieve relevant images for a given search term. An ontology is the "explicit specification of a conceptualization" (Gru95), and therefore characterizes entities and their relationships. The ontology can be used to detect semantic concepts represented by images (sections 3.1 and 3.2) as well as to create a tree of concepts related to the search term (section 3.3). Images are then assigned to nodes of the search tree according to their detected semantic concepts (section 3.4).

3.1 The WordNet Ontology

The official web page describes WordNet as a freely and publicly available "large lexical database of English nouns, verbs, adjectives and adverbs, grouped into sets of cognitive synonyms (*Synsets*), each expressing a distinct concept"[6]. That means, a Synset is a particular concept which can be expressed by different terms but has one unique identifier. The identifier consists of the word most commonly used to describe the concept, the part of speech, and a number, e.g. *drive.v.02*.
The number is necessary because one word can have multiple meanings that will then be represented by different synsets, like in *cherry.n.01* for the tree and *cherry.n.02* for the fruit. All Synsets a certain term may represent can be obtained by calling *wn.synsets("term")*. This call includes stemming the term, so its plural or conjugations will be matched as well.

Synsets are linked with each other through several semantic relations, e.g. *part-of*, *member-of* (meronyms) or *type-of* (hyponyms) relationships. In our work, we use this network of synsets to discover the semantics between terms describing the images as well as towards the search term.

Another popular ontology we could have used to explore semantic relationships is DBpedia[7], which is a Linked Data Project based on Wikipedia's infoboxes. Compared to WordNet, DBpedia contains more information in terms of entities, relationships, and attributes, and has support for multiple languages. However, since it is open data, it is by far not as well-structured as WordNet, and often inconsistent or redundant.

3.2 Assigning Keywords to Pictures

The first steps that had to be taken was to identify valuable image annotations, and to then find the terms' meanings in order to map them to the correct Synsets.

3.2.1 Annotation Data

We considered the following annotations provided by the Flickr API and evaluated them on twenty randomly sampled pictures:

[6] http://wordnet.princeton.edu/
[7] http://www.dbpedia.org/

- *Title.* The title was usually a short but precise description of the image content and thus very valuable for semantic annotation.
- *Description.* The description did often relate to the image content but with a lot of fill words and noise as well as context-dependent meanings, so it could be useful but would require additional preprocessing such as Named Entity Recognition.
- *Comments.* Only very few comments described the image in any way - they were mostly used for social interaction with the photographer.
- *Tags.* Tags are short, precise keywords on various abstraction levels. The vast majority of them are directly related to the image contents, and only little noise present due to the absence of fill words.
- *Album Names.* There are albums for diverse purposes, many of them related to the images' contents. Their names, however, tend to be obscured with special characters and the like, so quite some effort would be necessary in preprocessing.
- *Group Names.* The observations on group names were similar to those on albums.

Based on these findings, we use the single words from the title (split by whitespace) as well as tags. Before trying to find their corresponding Synsets, the keywords are cleansed: All those including digits are removed, since they more often represent image metadata (such as camera model, lens width, date, etc.) than information on the image contents. Additionally, all remaining keywords are stripped of special characters to achieve a more uniform representation. An endless number of additional filters could be introduced to avoid matching errors, but it must also be considered that potentially valuable information will also be removed by these filters.

3.2.2 Synset Detection

The difficulty in assigning Synsets to images is that there are multiple possible Synsets for a word, and it is obvious to a human observer but not to a computer which meaning is correct. Assuming that annotations on each image are closely related because they describe the same image content, we use those Synsets that, altogether, give the smallest semantic distance across all annotations of an image. Semantic distance of two terms can be measured by the length of the path between them in the WordNet tree. We use the Leacock and Chodorow Normalized Path Length (LCH-Similarity) provided by WordNet, which uses adapted weights and normalization factors, because it is perceived as closer to human understanding than regular path similarity (BH01).

To efficiently find the set of Synsets with the smallest overall distance, a best-first search algorithm [8] is used. Note that such search algorithms require non-negative distances between options, but WordNet provides similarities. To convert them into distances without

[8] Please refer to Artificial Intelligence literature, i.e. (Kum08) for a detailed explanation.

changing the scale, the similarity is simply subtracted from the maximally possible similarity, i.e. the similarity of a Synset to itself, which is roughly 3.7. For complexity reasons, only the best 100 candidates are considered at any time. Of course, this does not guarantee the perfect result anymore, but other paths are highly unlikely to become the best candidate in the end, and keeping all candidates would decrease performance significantly.
We also limit the matching to nouns, for two reasons: First, nouns are usually the words describing the depicted concepts. Second, the LCH-Similarity described above is only available within a part of speech(**?**).
This strategy provides decent results, although erroneous matching still occurs. One cause are words that are meant in a way that is unknown to WordNet, i.e. canon as the camera model might be interpreted as the type of music piece. Another cause are adjectives, adverbs and verbs that also exist in a noun form. The most common cause of this effect are pictures tagged with colors, because most terms describing a color also exist as nouns, like "orange" for the fruit, or "white" for a Caucasian person. We decided to add a filter to the preprocessing phase, so that all terms that can represent a color are removed.

Even with preprocessing, not all keywords can be matched to a Synset, because they are simply not represented in WordNet. The information about these *unmatched tags* is kept nevertheless, and later used for image retrieval, described in section 3.4.

3.3 Constructing a Search Tree

In general, all words represented in WordNet can be used as a query term for our tool. For the given use case, however, most query terms represent visible concepts like object descriptors at various levels of specificity, and place names. So our work is focused on these types of search terms.
When a term is entered into the tool, it is first used to retrieve all Synsets that can be expressed by this term. For each of them, a separate search tree is constructed, as can be seen in Figure 2, showing excerpts of the search trees for "bird" (*bird.n.01* and *bird.n.02*).

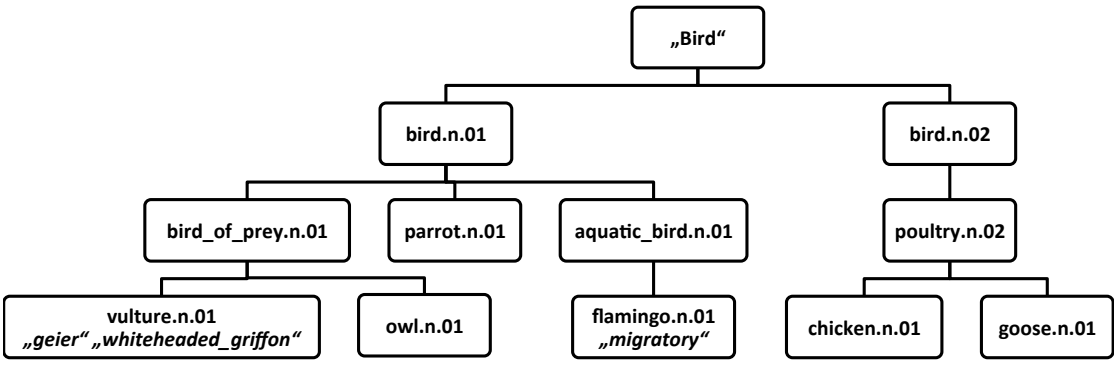

Figure 2: Examplary search tree (excerpt) for search term "bird"

The same figure also visualizes that a search tree is a tree of specializations. These specializations are retrieved using WordNet's hypnoym relations. For some terms, especially geographic Synsets, specializations are not applicable, so we use part-meronyms (part-of relationships), when no hyponyms are available.

Figure 3 shows the internal data structure of the tree. Each node represents one Synset, and references a list of more specific Synsets (*hyponyms* or *meronyms*).

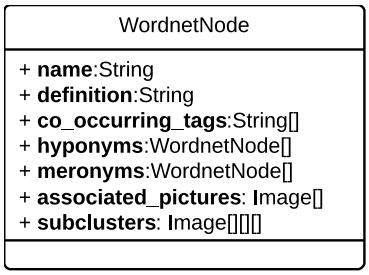

Figure 3: Tree node data structure

3.4 Assigning Pictures to Tree Nodes

Generally, assigning pictures to tree nodes is simple: Each node gets linked with all images that have been annotated with the Synset it represents during Synset detection.
In addition, strongly co-occurring tags are used for a higher recall. Co-occurring tags are those keywords that could not be matched to any Synset. They may, however, be closely related to certain Synsets, with which they often occur together. When that is the case, the keyword is added to the list of *co_ occurring_ tags* of the node.
We define a *strong* co-occurrence based on term frequency and inverse document frequency (tf-idf) values. The term frequency describes a normalized frequency of a term t_i in a document d_i:

$$tf(t_i, d_i) = \frac{freq(t_i, d_i)}{\max_k freq(f_k, d_i)}$$

In our case a Synset is regarded as document and an unmatched keyword as the term. The $freq(t_i, d_i)$ is the number of co-occurrences of that unmatched keyword and Synset. On the contrary, the inverse document frequency describes the inverted frequency of a term over all documents:

$$idf(t_i, D) = \log \frac{|D|}{|\{d \in D; t_i \text{ occurs in } d\}|}, \text{ with } D = \{d_1, ..., d_n\}$$

Unmatched keywords and Synsets again are regarded as terms and documents. Therefore, idf has a high value if an unmapped keyword occurs only with few Synsets. It is considered to be more important for the Synset than for example a camera model which occurs with many Synsets. Tf and idf are multiplied to consider both the number of occurrences with a certain Synset and the number of overall occurrences. If a simpler co-occurrence measure (e.g. the ratio of co-occurrences to the total number of occurrences

of the term) was used, very common keywords like camera models would be strong co-occurrences with many Synsets despite the lack of an actual relation.
We observed that the co-occurring keywords can be useful to find terms in foreign languages and proper nouns, but of course also introduces noise. The key to the quality of this features is the choice of the threshold. Reasonably good results were achieved with $0.75 * max\_tf\_idf$, where max_tf_idf is the maximal score across all values.

After adding all pictures that are annotated with the Synset itself or one of the related tags to the node's *associated_pictures*, some nodes may only have one or very few images. To create a balanced result with image sets of a significant size, nodes considered too small are merged into their parent node. Whether a node is too small is determined by the parameter *minimal_node_size*, which states the minimal number of images a node must have. To avoid merging of small nodes completely, the parameter should be set to zero.
The merge process is simple: All associated pictures of the node are combined with the parent node's pictures via union. Existing subnodes are not modified.

The above described steps of the Image Tree Creation phase are summarized in figure 4.

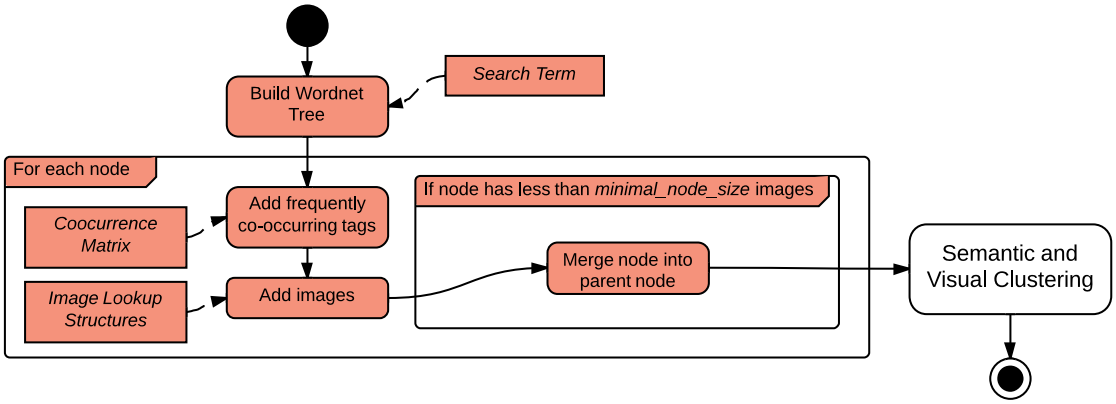

Figure 4: Process of Image Tree Creation

4 Semantic and Visual Clustering

The nodes received through the tree-based search are potentially very large (i.e., many pictures were found for the node). We found a rather small semantic and a large visual diversity within these nodes. It is therefore appropriate to refine especially the large nodes into smaller clusters.

4.1 General Approach

Since semantics are more meaningful to humans, and thus likely to be more important for the given use case, the refinement is done first on a semantic and then on a visual basis. That is, the results from the groups with semantically similar pictures are clustered again into subclusters with visually similar pictures. The steps are explained in more detail in sections 4.2 and 4.3. This approach has the additional advantage that outlier images, which have been assigned to a node but do not quite fit with the others because they show something different, can be filtered out in the semantic step.
The subclustering explained below and summarized in figure 5 will only take place for nodes/clusters with a certain minimum size and results in the structure of three nested Arrays of the WordnetNode class' attribute *subclusters* shown in figure 3.

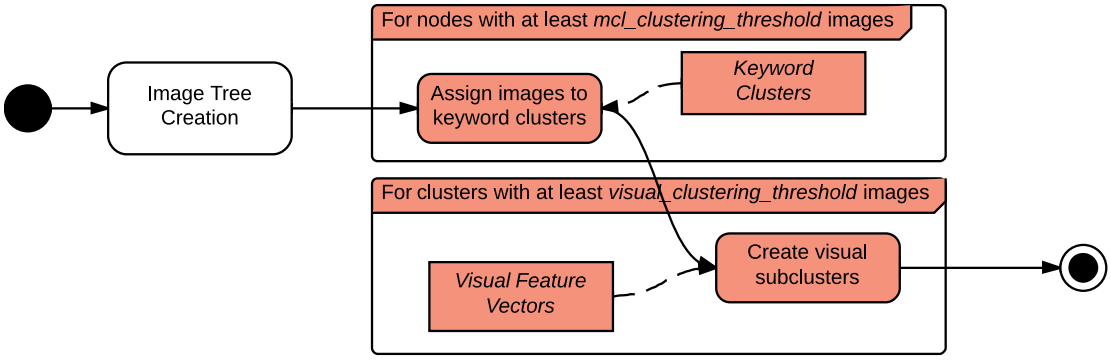

Figure 5: Process of Semantic and Visual Clustering

The data structures used in the process, such as Image Lookup Structures and Visual Feature Vectors, need only be calculated once for each image set. The preparational processes are visualized by figure 6.

4.2 Keyword Clusters

Semantic clustering is accomplished by using the associated Synsets which the Synset detection assigned to each image (see chapter 3.2.2). Therefore, Synsets are clustered into groups and images are assigned to these groups.

According to the paper "Automated Tag Clustering" by Grigory Begelman (BKS+06), our first approach of clustering Synsets used co-occurrences to span a graph of related Synsets. This graph consists of nodes representing the Synsets and edges representing

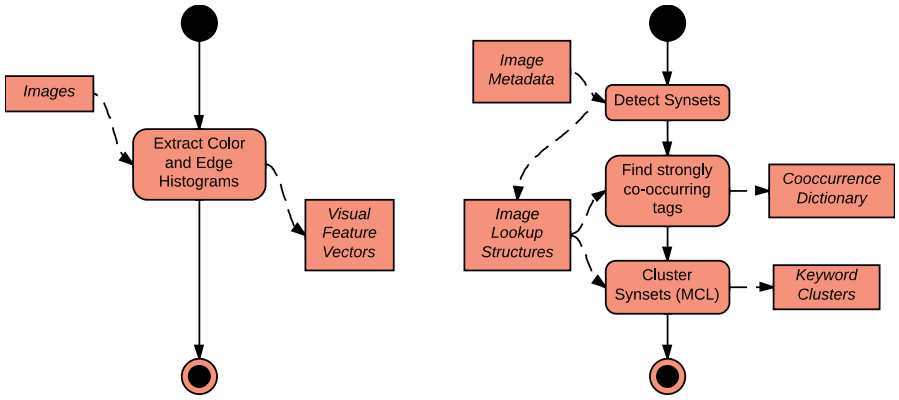

Figure 6: Static structures creation processes

the number of co-occurrences between Synsets. To include the advantages of Wordnet in the graph, we decided to replace the number of co-occurrences by a combination of co-occurrences and LCH-Similarity. The paper describes a graph clustering algorithm to achieve a grouping of related Synsets. But, the algorithm requires calculation of eigenvalues and eigenvectors for large sparse matrices. Furthermore, it does not take edge weighting into account. Consequently, we replace the graph clustering algorithm by the Markov Cluster Algorithm (MCL) introduced by Dongen (Don98). MCL is based on the Random Walk Model (Spi01). The basic idea is, if you start to walk from a node, it is more likely to stay inside a cluster than to leave it. Therefore, we calculate the probability to reach a node B from another node A in only one step. Then, we walk steps through the graph until the probabilities converge. The resulting probabilities inside a cluster are higher than outside. So, they can be used to determine groups of related Synsets.

To assign images to Synset clusters, we count how many Synsets the image shares with each cluster. It is then assigned to the cluster with the most matching Synsets. If two clusters have the same number of matching Synsets, the image is assigned to both clusters. As a consequence, a semantic cluster consists of images which have many associated Synsets in the same Synset cluster. For example, some pictures with parrots fall into a Synset cluster with persons, others in those with trees.

The advantages of an additional semantic clustering in each node of the image tree are obvious when looking at “Africa” as search term. The spanned tree consists of part-meronyms which are the countries of Africa. However, the pictures show people, animals, vegetation, cities, or other content not related solely to Africa. Simply with the help of the image tree, it is not possible to separate between those categories. Semantic clustering, allows in this context a more fine grained clustering. Also, in a well separated tree it is possible to achieve a more fine grained clustering. Pictures of the tree node *parrot.n.01* could be separated into pictures showing parrots in nature and pictures showing parrots in a zoo. Furthermore, semantic clustering permits the detection of outliers. If too few images fall into the same semantic cluster, they are considered to be outliers, and they are deleted from the tree node. For instance, a picture showing a cat whose name is

"Alexandria", which is a city in Africa, can be deleted from the tree for the search term "Africa".

4.3 Visual Clusters

One difficulty in the visual part of our work, besides the choice of appropriate features and their implementation, is the question how to use them jointly in a suitable algorithm for clustering.

4.3.1 Features

The features we chose for our tool are:

- Color histogram in HSV color space with 20 bins (i.e., 20 ranges) each
- Edge histogram length and angle histograms with 10 bins (i.e., 10 different angles considered) as combined vector

The reasons we chose these are that they are easy to calculate, rather obvious and humanly comprehensible. Since the purpose of this visual clustering is only in refining the semantic clusters, and not in trying to distinguish concepts by visual features, there is no apparent need for the use of more complex features.
For feature extraction, we use a pyramidal approach similar to the one proposed in (LSP06). Its advantage is that it combines features extracted over the entire image with features extracted on separate regions. The advantage of splitting images into regions for feature extraction is that, for example, two images with the same colors but in different structures will not automatically have the same color feature vectors (as visualized in figure 8). At the same time, the images' structure gains an unproportionally high importance, especially when using a large number of regions. When using 5x5 rectangles, for example, it can be observed that images with same-colored borders were considered very similar, independent of their actual content.

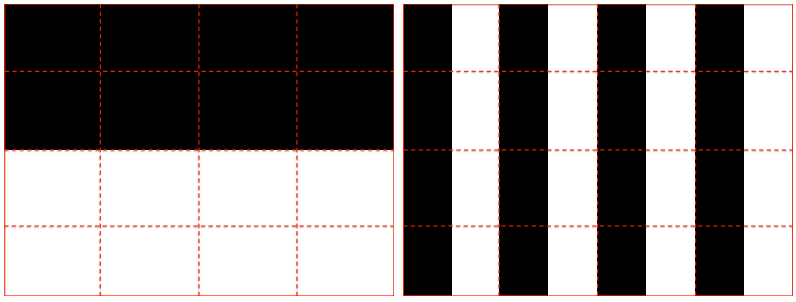

Figure 7: Two images that have the same color histogram regarding the entire picture but different histograms on the marked regions

With the applied pyramid technique, the final feature vector is concatenated from several partial feature vectors, labeled from 1 to 30 in figure 8. The appropriateness of this method especially for refining existing clusters, is also discussed in (LSP06).

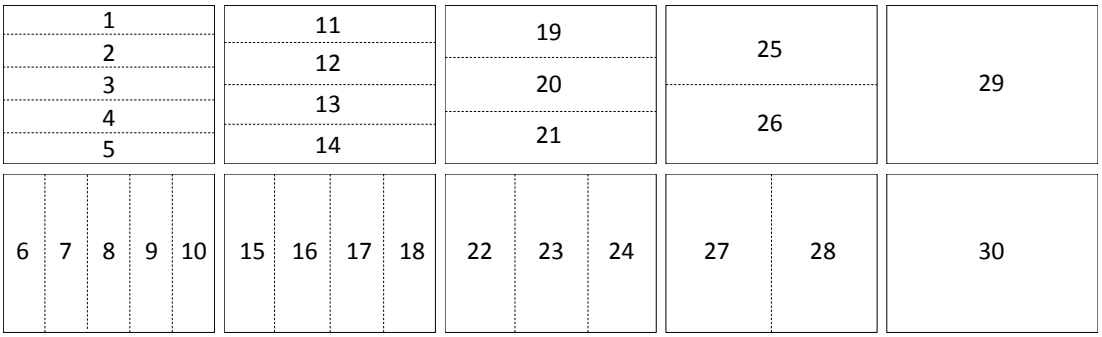

Figure 8: Pyramidal image splitting for feature extraction

4.3.2 Clustering

A first, rather naive approach to clustering the visual characteristics extracted would be to concatenate the feature vectors (histograms), and apply one of the established clustering algorithms such as k-means. The fact that remains unseen in this approach is that, generally, the values of different features are usually measured on different scales and therefore vary in their orders of magnitude. They also consist of a different number of dimensions.
These circumstances influence any algorithm based on the distance between two images. Since differences in the larger values will usually be larger in its absolute value, they will also be more influential to the overall distance than the dimensions with smaller values. Furthermore, the feature with more dimensions will always be more influential.

So, instead, k-means is applied separately for colors and edges, and the results are later joined through *late fusion*, as explained below. As no specific criteria exist for the number of clusters that should be achieved, k is chosen by the established rule of thumb: $k = \sqrt{n/2}$ (MKB79, p.365), where n is the number of items to be clustered. K-means was chosen over hierarchical clustering, because it provided more well- and equally-sized clusters, the latter often just split off single images.
Initially, we planned to use an adaptive k, that is, start with a small k and increase it until the error (mean distance from centroids). Despite its higher computation complexity, it does not seem to provide better results than the rule of thumb. For example in color clustering, the adaptive approach often just separates black and white images from colored ones.
We combine the single-feature clusters by intersecting them, which is a simple and performant late fusion method . It ensures that all images within a cluster are similar in color as well as edge structure and leads to less or equal to $n/2$ subclusters.

5 Evaluation

We evaluated our tool on a set of 9,201 images, which are a subset of the MIRFLICKR-1M file set, and the query term "food". Since no comparable algorithms exist, the evaluation is mainly aimed at obtaining the best values for the parameters and at providing a basis for comparison of further improvements and future work.

5.1 Test set

No gold standard is available to tell us which pictures show food and how similar the images are. The creation of such standards and training data is exactly the task we want to facilitate with this work.
What we did to receive evaluation data was to crowdsource the needed data from the general public. This was achieved in two phases:

First, the users were shown random picture out of the 9,201 test set images and asked whether it shows food or not. We normalized these answers, so that there is only one vote per user per picture. In the case a user rated a picture more than once, the value is determined by the ration of positive ("shows food") and negative ("does not show food") votes of that user on that picture. We consider all those images as showing food that received at least 50% positive votes. With over 35,000 clicks by more than 40 participants, 1,142 images out of the total 9,201 images were identified to show food.

We also need data on the semantic and visual similarity of the pictures. Therefore, in the second phase, the users were shown pairs of images and asked to compare them. They could choose between three levels of semantic similarity: *not similar*, *same object*, and *same object and same context*, and two levels of visual similarity: *similar*, and *not similar*.
Among the 12,962 votes were 771 pairs of images with same objects, 354 pairs with same object and same context, as well as 1,885 pairs of visually similar images. The same normalization as in the first phase was applied.

5.2 Quality Indicators

The evaluation focusses on the following four main aspects of our algorithm:

1. Retrieval of matching images
2. Semantic hierarchy and clusters
3. Visual clustering

should we compare synset detection mechanisms?

We measure the quality of the image retrieval (1.) by precision and recall of returned pictures, compared to those that were declared to show food by the test persons.
The quality of the hierarchy of the retrieved images (2.) is based on the same object and same object and context pairs: The minimal path distance for an annotated pair of pictures can be calculated and used to determine the closeness of two images

$closeness(x, y) = 1/distance(x, y)$. Averaging this value over all pairs of a similarity category returns a value between 0 and 1 (below referenced as c_o for same object pairs, c_c for same object and same context pairs, and c_n for not similar pairs), with the optimal values being 1 for positive (similar) pairs, and 0 for negative (non-similar) pairs. We include the keyword clustering in this evaluation by handling the clusters in a node as its children. Consequentially, the perfect score of 1 can only be reached when to semantically similar images are not only in the same node but also in the same semantic subcluster.
Visual similarity (3.) is evaluated on the whole testset, because not enough comparison data is available to get valuable results if only comparisons within semantic clusters were used. Once again, precision and recall are used as indicators.
vary parameters given by frontend, trying to find best configuration

5.3 Results

5.3.1 Phase 1

Our image retrieval has a precision of p = 50.2% and recall of r = 85.9% on the "food" query, before execution of the semantic clustering that removes outliers. Without the use of co-occurring tags described in section 3.4, both values show no significant difference with p = 50.5% and r = 85.4%. After the semantic clustering, the measures depend on the *minimal mcl cluster size* parameter. The results for different values of this parameter are presented in table 1.

5.3.2 Phase 2

Within our second phase we let users decide which of the pictures declared as showing food are semantically and visually similar.
Image tuples voted with same object: 757
Image tuples voted with same object and same context: 1102
Image tuples voted as visual similar: 1854

After retrieving our evaluation votes from the users, we compared the result of our tool with one from the evaluation. Therefore we calculated the distance between every two images, which are declared as similar or not, within our calculated search tree.
Average distance for same object is 3.70286885246
Average distance for same object and same context is 3.54583921016
Average distance for not similar is 3.81995379729

Picture of a search tree, which illustrates the evaluation script concerning how the distance is calculated?

mcl clustering threshold	*minimal mcl cluster size*	*minimal node size*	*precision*	*recall*	*f-measure*
0	0	0	0.501532	0.859143	0.633344
2	6	2	0.575916	0.769904	0.658929
4	2	2	0.518719	0.836395	0.640322
4	2	4	0.520458	0.834646	0.641129
4	6	2	0.562267	0.790026	0.656966
4	4	4	0.550663	0.798775	0.651910
4	6	4	0.568766	0.774278	0.655798
6	6	6	0.567619	0.782152	0.657837
5	10	5	0.612903	0.748031	0.673759
5	15	5	0.644391	0.708661	0.675000
5	10	10	0.598080	0.762905	0.670511
10	10	10	0.598080	0.762905	0.670511
10	10	20	0.578165	0.783027	0.665180
10	20	20	0.642631	0.709536	0.674428
10	15	10	0.631539	0.728784	0.676686
10	20	10	0.666102	0.687664	0.676711
15	20	15	0.661716	0.701662	0.681104
15	25	15	0.695298	0.672791	0.683859
20	20	20	0.642631	0.709536	0.674428
20	25	20	0.677951	0.683290	0.680610
50	50	50	0.728135	0.604549	0.660612
100	100	100	0.757858	0.569554	0.650350

Table 1: Precision and recall of the image retrieval

6 Results Discussion

Are our results good? Are they biased by something?

It can generally be said that the quality of the results highly depends on the original image annotations: an inappropriate tag leads the algorithm to "believe" that the picture shows something that is actually not present.

6.1 Testset Quality

The evaluation results also depend on the test set, which, unfortunately, cannot be clearly right or wrong. Different users will expect different images to be returned according to their defintion of food: When some of the participants of the test set creation were asked which items they considered food, the answers ranged from "Those that I would like to eat" to "Anything that some living organism would eat".

how many users?

It also has to be assumed that people have different opinions on what images are visually similar, especially since no definition or hints were given to the participants. We used crowdsourcing to deal with these problems and obtain a test set that is supported by the majority of users. So the key question to the quality of the test set is whether participants

?	*minimal node size*	c_o	c_c	c_n
value 1	value 1	value 1	value 1	value 1
value 2	value 2	value 2	value 2	value 2
value 3	value 3	value 3	value 3	value 3

Table 2: Semantic quality measures

are enough to obtain a representative result.

6.2 Image Retrieval

One of the reasons for the generally poor precision of the image retrieval may lie in poorly annotated images. Other, more controllable reasons, whatsoever, are to be searched in the Synset detection mechanism. First, the limitation to nouns leads to incorrectly identified Synsets, because adjectives, adverbs and verbs are wrongly matched to nouns if such exist. Second, words in other languages than English may be incorrectly matched if they exist in a different meaning in English., And third,

Examples!

6.3 Semantic Clusters

MCL based clusters highly depend on quality of keyword clusters. Hard to evaluate, cannot be isolated.
Other problems during test set creation include the fact that pictures often contain small or processed items, which makes it hard to identify the exact contents of that picture. The original tags therefore may contain more or contrary information to what the participants could see.

phrase this better or remove

6.4 Visual Clusters

also rather hard to look at in isolation, because method specifically designed for final subclustering. But lack of data for evaluation within subclusters for appropriately sized semantic clusters

7 Future Work

How to improve, what other approaches to take

7.1 Semantic

use more or other WordNet relations
improve keyword clusters by re-clustering large clusters
better synset detection (still see faulty recognition of tags), use groups and albums additionally, description with named-entity recognition

7.2 Visuals

A Glossary

Late Fusion:

Synset:

WordNet:

Markov Clustering Algorithm: a graph clustering algorithm for undirected, weighted graphs using random walk to determine clusters

Leacock and Chodorow Similarity:

B Abbreviations and Acronyms

Bsp	Beispiel
LCH	Leacock and Chodorow
MCL	Markov Cluster Algorithm

References

[BH01] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *In Workshop On Wordnet And Other Lexical Resources, Second Meeting Of The North American Chapter Of The Association For Computational Linguistics*, 2001.

[BKS+06] Grigory Begelman, Philipp Keller, Frank Smadja, et al. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, pages 15–33, 2006.

[CHL+04] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of www image search results using visual, textual and link information. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 952–959. ACM, 2004.

[Don98] Stijn Van Dongen. A new cluster algorithm for graphs. Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, 1998.

[Gru95] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928, 1995.

[Kum08] E. Kumar. *Artificial Intelligence*. I.K. International Publishing House Pvt. Limited, 2008.

[LKI11] Pei-Chin Lim, Narayanan Kulathuramaiyer, and Dayang NurFatimah Awg. Iskandar. Towards semantic clustering - a brief overview. *International Journal of Image Processing*, 4(6):556 – 565, 2011.

[LMS+09] Stefanie Lindstaedt, Roland Mörzinger, Robert Sorschag, Viktoria Pammer, and Georg Thallinger. Automatic image annotation using visual content and folksonomies. *Multimedia Tools Appl.*, 42(1):97–113, March 2009.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178, 2006.

[LZLM07] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262 – 282, 2007.

[MKB79] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[OW10] Razvan Orendovici and James Z. Wang. Training data collection system for a learning-based photographic aesthetic quality inference engine. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1575–1578, New York, NY, USA, 2010. ACM.

[RBV+11] Joni Radelaar, Aart-Jan Boor, Damir Vandic, Jan-Willem Dam, Frederik Hogenboom, and Flavius Frasincar. Improving the exploration of tag spaces using automated tag clustering. In S"oren Auer, Oscar DÃaz, and GeorgeA. Papadopoulos, editors, *Web Engineering*, volume 6757 of *Lecture Notes in Computer Science*, pages 274–288. Springer Berlin Heidelberg, 2011.

[RHMGM09] Daniel Ramage, Paul Heymann, Christopher D. Manning, and Hector Garcia-Molina. Clustering the tagged web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 54–63, New York, NY, USA, 2009. ACM.

[Spi01] Frank Spitzer. *Principles of random walk*, volume 34. Springer, 2001.

[ZIL12] Dengsheng Zhang, Md. Monirul Islam, and Guojun Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346 – 362, 2012.

Index