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.

Contents

Todo list

1 Retrieving Images in Clusters

introduce the introduction?

1.1 Problem Statement & Motivation

training data for image categorization and content detection
flickr and other online photo communities are good sources for annotated images
problems: low annotation quality, only search for specific term (with different meanings and visual characteristics)
for example, want to test the quality of my algorithm in identifying different foods: would have to think of all kinds of food, then search images and group them into homogeneous groups manually

What do we do?
clustering: creating homogeneous groups of semantically and visually similar pictures

Why do we do that?
seminar challenge: cluster 1 million pictures of the MIR1M flickr file set
improving the complex task of searching for pictures according to a given keyword
facing different challenges like: multiple meanings of the keyword, bad picture annotations, taking semantic and visual information of a picture into account

what is flickr? folksonomy etc.

1.2 Clustered Tree Nodes Approach

We implemented a web application in Python using SimpleCV[1] for visual image analysis and Flask[2] for the frontend.
The tool provides ready-to-use semantically and visually homogeneous image clusters for a given topic. This is achieved by first spanning a tree of subordinate terms, retrieving related images by their keywords for each term, and then clustering the images by their predominant keywords as well as by colors and edge structure. These two major phases are illustrated in figure 1

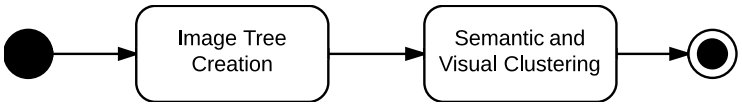

Figure 1: The two main phases of our algorithm

After giving an overview of Related Work in chapter 2, we will present how we analyze the image annotations and the user's search term to retrieve relevant images (chapter 3). Our methods to cluster these semantically and visually are described in chapter 4.

[1] http://www.simplecv.org
[2] http://flask.pocoo.org/

Chapter 5 explains how we evaluate our approach, while the evaluation results will be 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 these works present new algorithms for one of the above use cases, not methods to retrieve training data.

Related Subjects: Image Annotation, semantic clustering, content-based image retrieval

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).

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 is (LMS$^+$09), which tries to annotate images (with a defined vocabulary??) based on visual features and existing tags, so-called *folksonomies*. Their goal, however, is to create additional annotations for not or poorly tagged images.

3 Image Tree Based on WordNet

intro to chapter 3

3.1 WordNet

The offical web page [3] 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". That is, 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 that may be represented by a certain term can be obtained by calling *wn.synsets("term")*. This includes stemming of the term, so also its plural or conjugations will be matched.

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.

A popular alternative ontology we could have used to explore semantic relationships is DBpedia, which is a Linked Data Project based on Wikipedia's infoboxes[4]. multiple languages, but open data: not as well-structured and consistent, more information but strongly varying levels of detail.

Why did we chose WordNet over DBpedia?

3.2 Assigning Keywords to Pictures

The first step now is to identify valuable image annotations and associate them with their meanings, i.e. find the Synsets they represent.

3.2.1 Annotation Data

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

- *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.

[3] http://wordnet.princeton.edu/
[4] http://www.dbpedia.org

- *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 obscurred 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, and try to find the corresponding Synset. Before that, though, the keywords are cleansed: All those including digits are removed, since they more often represent image metadata (such as camera model, lense 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 [5] is used. Note that such search algorithms require non-negative distances between options, but WordNet provides similarities. To convert them into distances without 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

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

describing the depicted concepts. Second, the LCH-Similarity described above is only available within a part of speech. reference?
This strategy provides decent results, although erroneous matching still occurrs. 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 Searchtree

In general, all words represented in WordNet can be used as a query term for our tool. For the given use case, however, queries will be limited to those that can be seen in pictures. This is mainly the case for 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 searchtree is constructed, as can be seen in Figure 2, showing excerpts of the searchtrees for "bird" (*bird.n.01* and *bird.n.02*).

Find better description of what search terms we expect

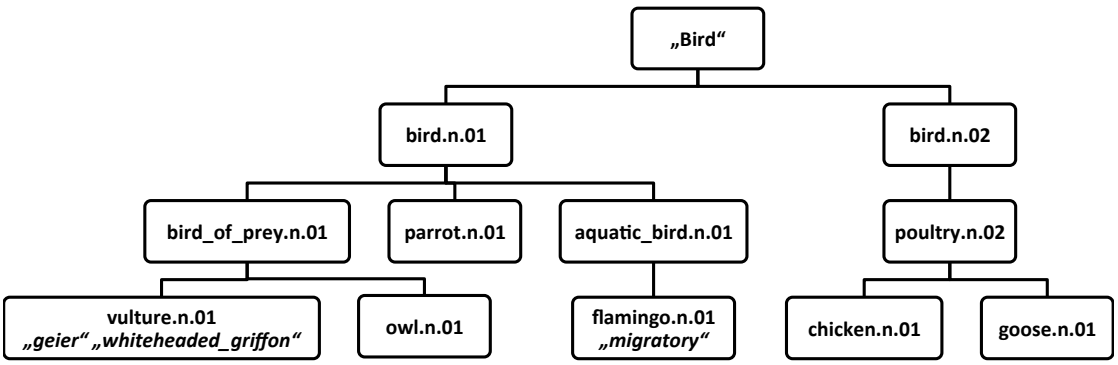

Figure 2: Examplary searchtree (excerpt) for search term "bird"

The same figure also visualizes that a searchtree 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*).

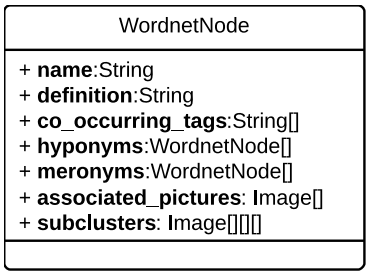

Figure 3: Tree node data structure

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 mapped to any Synset. They may, however, be closely related to certain Synsets, with which they often occurr 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 tf-idf values. 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.

explain tf-idf

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 0.
The merge process is simple: All associated pictures of the node are unioned into the parent node's pictures and the node itself deleted.
The above described steps of the Image Tree Creation phase are summarized in figure 4.

What happens when subnodes exist?

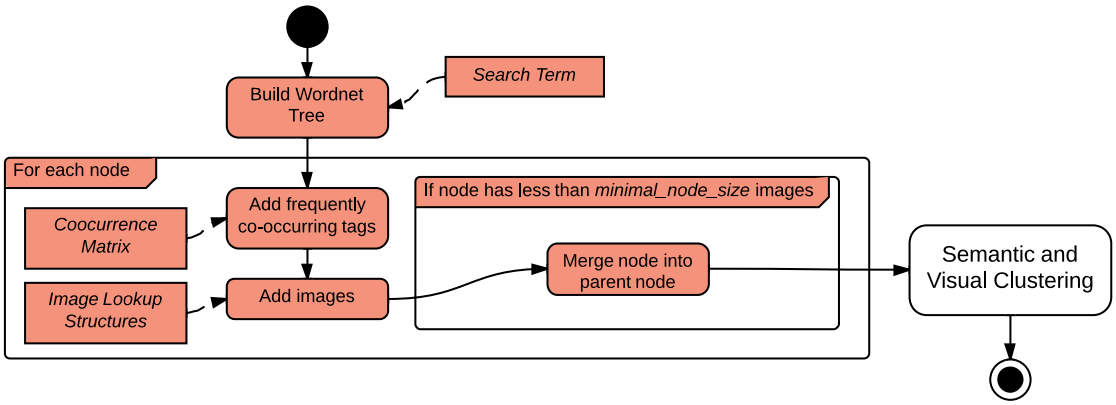

Figure 4: Process of Image Tree Creation

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.
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 assigned Synsets. Therefore, Synsets are clustered into groups and pictures are assigned to these groups. According to the paper of 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 the number of co-occurrences between Synsets. To include the advantages of Wordnet, we decided to combine the number of co-occurrences

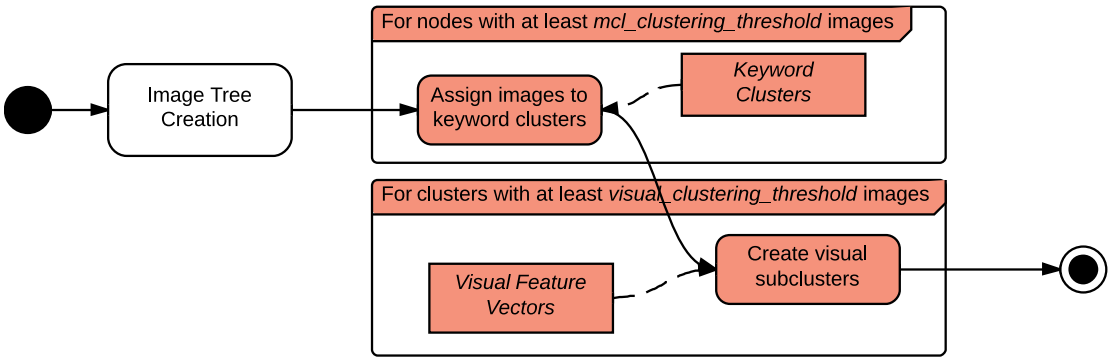

Figure 5: Process of Semantic and Visual Clustering

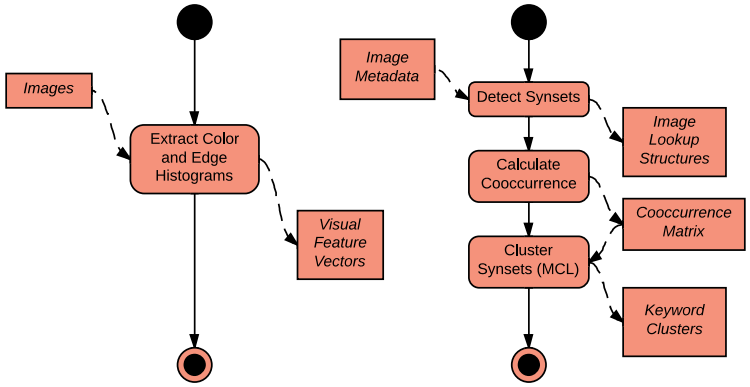

Figure 6: Static structures creation processes

with the LCH-similarity The paper describes a graph clustering algorithm to group 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, the graph clustering algorithm is replaced 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 probable to stay inside a cluster than to leave it. Therefore, we calculate the probability to reach another node B from a node A in only one step and then walk steps until convergence of probabilities. The resulting probabilities inside a cluster are higher than outside. So, they can be used to determine groups of related Synsets.

for each image, count how many synsets it shares with each cluster, and assign it to maximum (can be multiple), so all images with maximum tags in that keyword cluster

form one semantic cluster. Example: some parrot pictures fall into keyword cluster with persons, others in those with trees.
if only one image falls into a keyword cluster, consider it an outlier and remove it good for context(?), outlier identification, basic clustering for part-meronym spanned trees (Africa example)

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 to use in our tool are:

- Color histogram in HSV color space with 20 bins each
- Edge histogram lengths and angles, histograms with 10 bins (i.e., separation into 10 angles, with count of edges and sum of lengths for each) as combined vector
  For edge histogranm extraction we use the EdgeHistogramFeatureExtractor class from the SimpleCV library. Referring to the official SimpleCV documentation the method creates histograms for the edge lengths and angles. The number of bins is used to define which and how many line directions are taken in consideration.

make the following sound less copied

reference: http://simplecv.

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.

Can we explain or prove that somehow?

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 like 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: In color histogram, each bin's value represents a number of pixels, whereas in edge histograms the number of edges is counted, which is significantly smaller.
This circumstances influences 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. So, instead, we decided to apply k-means separately for colors and edges, and join the results later 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 provides no better results than the rule of thumb. For example in color clustering, the adaptive approach will often just separate black and white images from colored ones.
For feature extraction, we use a pyramidal appraoch similar to the one proposed in (LSP06). Its advantage is that ...
Same paper also states the appropriateness of this method especially in refining existing clusters.
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.

Explain advantages of pyramidal feature extraction

5 Evaluation

We evaluated our tool on a set of 9,201 images 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.

exact number?

5.1 Testset

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. We normalized these answers, so that there is only one vote per user per picture, shoe 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 users, 1,142 images were identified to show food.

how many users?

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 contex, as well as 1,885 pairs of visually similar images. The same normalization as in the first phase was applied.

5.2 Evaluation Method

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

1. Retrieval of matching images
2. Semantic hierarchy of retrieved images
3. Semantic clustering
4. Visual clustering

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

should we compare synset detection mechanisms?

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, with the optimal values being 1 for positive (similar) pairs, and 0 for negative (non-similar) pairs.
The same object and same context annotations can also be used to examine the quality of the keyword clusters (3.). (compare both, what does mcl actually do?)

how do we actually evaluate mcls?

We evaluate visual similarity (4.) on the whole testset, because not enough comparison data to get valuable results if we only use comparisons within semantic clusters. Calculate precision and recall

vary parameters given by frontend, trying to find best configuration

5.3 Results

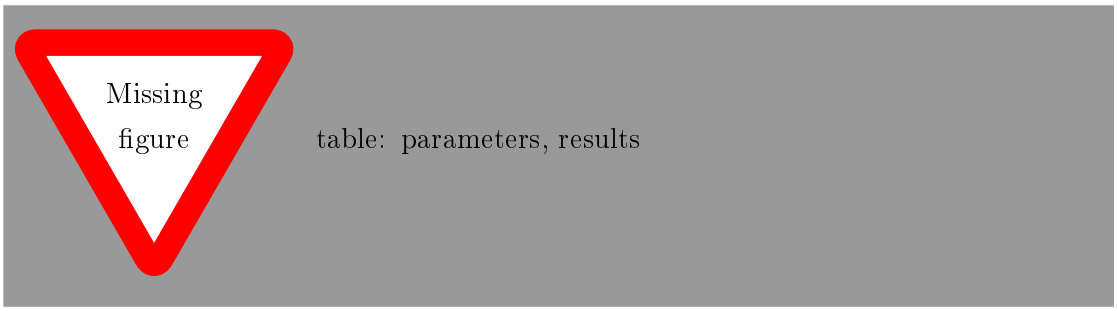

table: parameters, results

6 Results Discussion

All depends on annotations - inappropriate tagging leads to bad results, as well as limitation to nouns (adjectives, adverbs and verbs are wrongly matched nouns)

Are our results good? Are they biased by something?

6.1 Testset Quality

The evaluation results 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".
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 are enough to obtain a representative result.

how many users?

6.2 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

6.3 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:

Beispiel: eine Beispiel-Erklärung

Markov Clustering Algorithm:

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.

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

[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.

[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