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 the abstract

tool for collecting training data for semantic image analysis
images from folksonomies (user annotated) with tag information and title and comments
therefore using meta information for semantic analysis and image itself for visual analysis
With the help of WordNet, create a hierarchy of hyponyms and meronyms
each hierarchy element containing images are additionally clustered according to semantic meta information
and after that also according to visual information
evaluation with crowdsourcing
issues with test set, definition of semantic clusters and only basic visual analysis

rework titles

Contents

Todo list

delete list of todos

1 Retrieving Images in Clusters

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 with no existing control mechanisms. Secondly, one can only search for a specific term, and will obtain images for all of the term's meanings. However, this way the retrieval is limited to only those images which are annotated with the exact term as a tag. This results in the fact, that usually no semantic 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. 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.

1.1 Clustered Tree Nodes Approach

The tool has been implemented as a Python web application, using WordNet[2] for semantic image analysis, SimpleCV[3] for visual image analysis and Flask[4] together with Bootstrap[5] for the frontend presentation.
It provides ready-to-use semantically and visually homogeneous image clusters for a

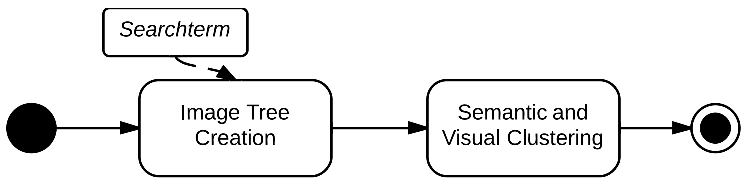

Figure 1: The two main phases of the clustered image search

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

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

1.2 Web Interface

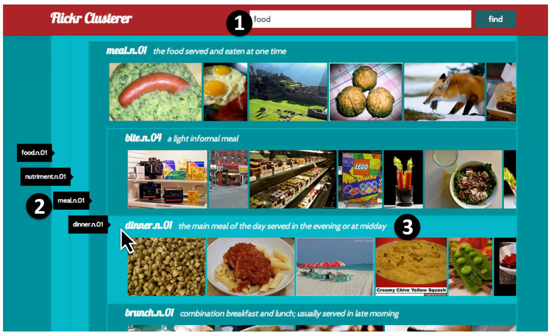

Figure 2: Overview of the web interface (1) search term entered by user (2) semantic hierarchy (3) each line represents one hyponym of the entered search term with associated images

Figure 3: Details view for one hyponym (1) selected hyponym with definition (2) dark cyan background groups different semantic clusters (3) white background groups visual clusters

After users entered a search term on the web interface (figure 2.1), they will be presented with a result tree. Each meaning WordNet finds for the given search term will result in a root node. Hyponyms of each node will then be rendered as sub nodes. This continues, until no more hyponyms can be found. As figure 2.2 shows, this hierarchy can then also be seen when hovering over the image rows. For each row, the users now see a preview of the images found for this particular meaning of their search term (2.3).

could be more comprehensive for chapter 1

If one search term now strikes the user's attention, they can get detailed information by clicking on the images. Figure 3 shows the *details view*, where the definition of the particular hyponym can be found (3.1). In the details view, the images now appear in more fine grained clusters. The dark cyan background groups semantically similar images (3.2) which have many other tags in common. These most dominant tags are also shown for each semantic sub cluster. Within the semantic clusters, the white containers group visually similar images (3.3). It is in these containers that users can find visually similar images belonging to one semantic topic of the given search term.

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 Flickr photograph quality. Comments are used to extract terms and factors which describe image quality. The actual analysis of the images and selection of training data, however, is done by humans, using a specialized voting tool also presented in their work.

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 most significant difference is that documents such as web pages consist of words, so their content itself can easily be used for semantic analysis.
Current issues with tag-based search and clustering, for both documents and images, 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. (RSM94) is a general introduction into the use of the WordNet knowledge base for dealing with common problems of the natural language. Without any specific use case, they identify the need for a disambiguator which automatically and correctly assign words to their WordNet meaning. Such a tool is claimed to be indispensable for all classification approaches. It is investigated how a word's meaning can be found. The presented technique spans a so-called hierarchical concept graph for a specific entity, consisting of its hyponymns and meronyms. The hierarchical concept graph is used to determine the semantic similarity between different words.
The results of the presented semantic tagger are promising and support our decision to use the advantages of WordNet within our tool, e.g. to map tags of an image to their correct meaning, in relation to other tags of the image.

WordNet has also been used as a foundation for a clustering technique for text documents, which is described in (SK04). The technique aims at solving the problem of synonymy and ambiguity of words within texts, while adding a part-of-speech tag to every word based on knowledge provided by WordNet. Their conclusion is that including synonyms and hypernyms does not improve the effectiveness of clustering, which is supposedly related to noise introduced by incorrect word interpretations when mapping terms to WordNet.

2.2 Automated Image Annotation

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.
Often the automatic annotation of images is based on the existence of a training set of previously annotated images. (JLM03) uses an approach to annotate images with the help of blobs generated from an image. The idea using formal information to tag images with their correct meaning differs from our basic idea to rely on the given metadata, annotated by users from Flickr.

Another approach is presented in (WDS$^+$01), where user feedback and automatic image annotation are combined in order to become both accurate and at the same time efficient. The idea is based on the automatic assignment of keywords for an image which receive positive user feedback. It is integrated into search mechanisms, where images that well fit to the user's query can be annotated with the query term. This is a way to face the problems which come with manual annotation (low efficiency) and automatic annotation (low accuracy).

2.3 Combination of Semantic and Visual Approaches

One work that has used the idea of combining semantics and visuals to analyze pictures in a so called *visual folksonomy* is (LMS$^+$09). The idea is trying to annotate images, with a controlled vocabulary, based on visual features and existing tags. Their goal, however, is to create additional annotations for not or poorly tagged images.
Similar ideas are presented by (CHL$^+$04) with the aim to cluster images returned by a web image search. In contrast to pictures from a folksonomy, image search results are taken from "regular" web pages and connected with surrounding context and link information. This information is then used by their algorithm to cluster images and provide a more well-organized result overview.

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(PPM04).
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 4, showing excerpts of the search trees for "bird" (*bird.n.01* and *bird.n.02*).

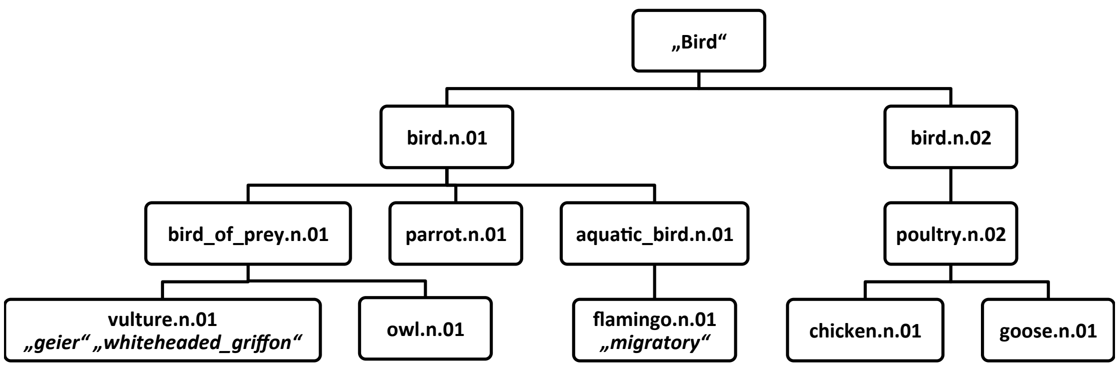

Figure 4: Examplary search tree (excerpt) for search term "bird"

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

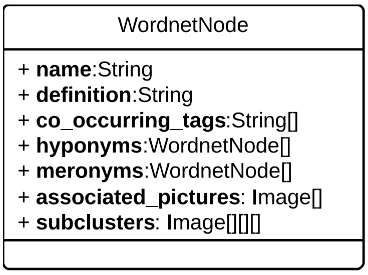

Figure 5: Tree node data structure

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 6.

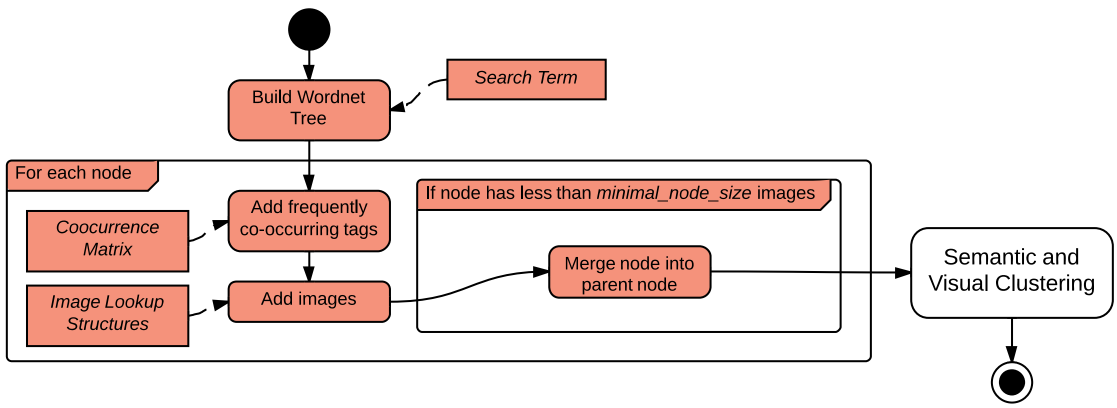

Figure 6: Process of Image Tree Creation

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 7 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 5.

add Search Term to graphic

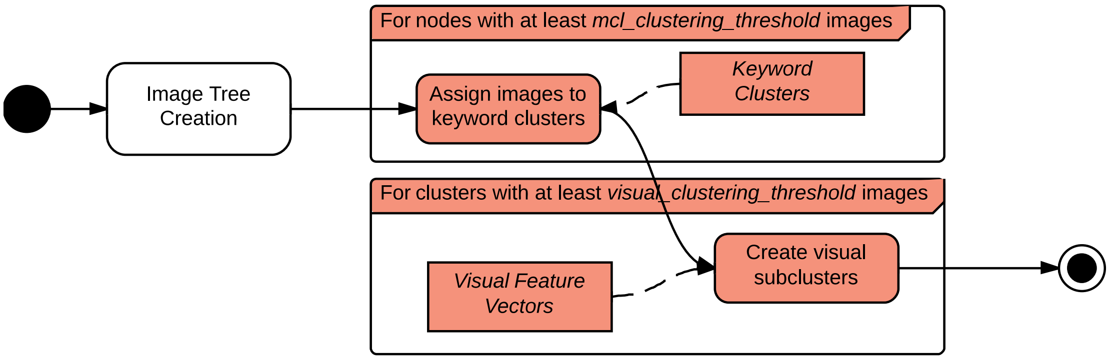

Figure 7: Process of Semantic and Visual Clustering

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 8.

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). Therefor, Synsets are clustered into groups and images are assigned to these groups.

The paper "Automated Tag Clustering" by Grigory Begelman (BKS+06) presents an algorithm for tag clustering based on graph clustering. It uses co-occurrences of tags to

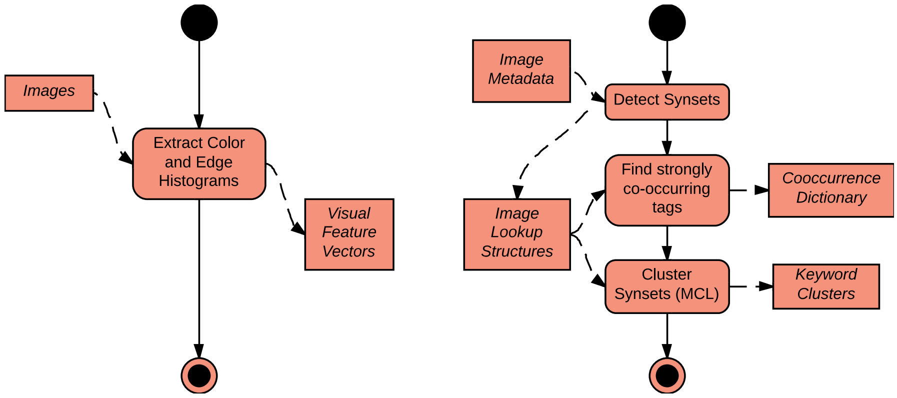

Figure 8: Static structures creation processes

span a graph whose nodes represent tags and whose edges represent co-occurrences. We adapt the algorithm to include the advantages of WordNet. That is why, we replace the number of co-occurrences by a combination of co-occurrences and LCH-Similarity. A disadvantage of this algorithm is the graph clustering via eigenvalues and eigenvectors that is expensive for large graphs. Furthermore, this algorithm 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 9). 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.

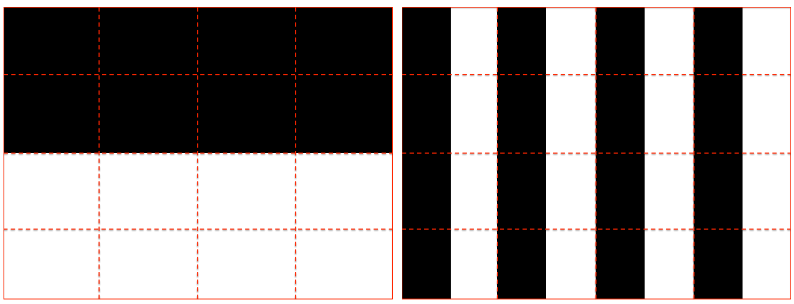

Figure 9: Two images that have the same color histogram regarding the entire picture but different histograms on the marked regions

With the applied pyramid technique, the final feature vector is concatenated from several partial feature vectors, labeled from 1 to 30 in figure 9. The appropriateness of this method especially for refining existing clusters, is also discussed in (LSP06).

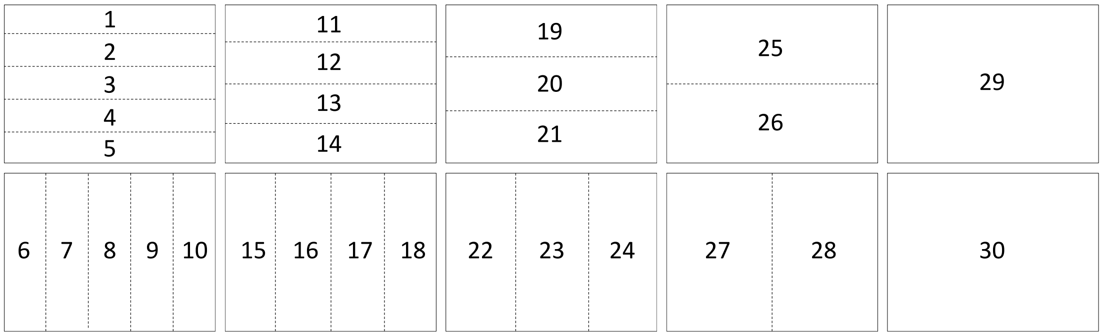

Figure 10: Pyramidal image splitting for feature extraction

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 which is a recursive k-means approach (dHIM05, p.17-20), because it provided more well- and equally-sized clusters, the latter in hierarchical clustering often just splits 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) no longer decreases. 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 and the query term "food". The images are a subset of the 1 million images of the MIRFLICKR-1M[9] file set. 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.
To test the quality of our algorithm, we wrote a tool allowing us a crowdsourced generation of the needed reference data by the general public. This was achieved in two phases:

First, the users were shown a 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 multiple times, the value is determined by the ratio of positive (*"shows food"*) and negative (*"does not show food"*) votes of each user on one picture. We consider all those images as showing food that received more than 50% positive votes. With over 35,000 clicks by more than 20 participants, 1,142 images out of the total 9,201 images were identified to show food.

Since data on the semantic and visual similarity of these pictures is also necessary for the evaluation, in the second phase, the users were shown pairs of images and were asked to compare them. Those images were, in the first phase, claimed to contain food. Users 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, of more than 30 participants, were 757 pairs of images with same objects, 345 pairs with same object and same context, as well as 1,854 pairs of visually similar images. Multiple votes on one pair were rare (39 cases), and therefore simply not taken into account if they contradicted each other.

5.2 Quality Indicators

The evaluation focuses on the following three main aspects of our algorithm:

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

[9] http://press.liacs.nl/mirflickr/

We measure the quality of the image retrieval (*1.*) by calculating the F-Measure for the returned pictures, comparing our algorithm's result to the crowdsourced generated test set of the first phase.
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. Optimal values are 1 for positive (similar) pairs, and 0 for negative (non-similar) pairs. The similarity categories are 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. 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 two semantically similar images are not only in the same node but also in the same semantic subcluster.

As described in chapter 4.3, we divide the already formed semantic clusters again into smaller visual clusters. To test the performance of this visual subclustering (*3.*), we compared the results of our algorithm again to the test set from the second phase of the crowdsourced generation. Precision and recall are calculated by analysing how many images, which are annotated as *visually not similar*, are actually separated in different visual clusters. The test set contains 10,894 associations annotated as *visually not similar* and 1,841 as *visually similar* annotated image pairs. Since there are approximately 1.3 million possible combinations between two images within the images annotated as food, this is an approximate 1% coverage, meaning we know only for every 100th combination, if or if not it is considered visually similar. Therefore, it was not possible to gain any valuable information on the subclustering we do within the semantic clusters.
For being able to, anyhow, test the performance of our visual clustering, we decided to perform this clustering on the retrieved images without semantic clustering. Therefore, we filtered the results, our algorithm provides for the search term "food", and took only those into account, which users voted to contain food. 883 images remained. This way, we prevent to visually cluster irrelevant images. Since, our clustering algorithm for visual features uses k-means which is based on a random distribution, we performed multiple measurements to reduce the error.

5.3 Results

5.3.1 Image Retrieval

Our image retrieval has a precision of 50.2% and recall of 85.9% on the query "food", 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 parameters *minimal node size*, *mcl clustering threshold*, and *minimal mcl cluster size*, described in sections 3.4 and 4.2. The results for different values of this parameters are presented in table 1.

mcl clustering threshold	*minimal mcl cluster size*	*minimal node size*	*precision*	*recall*	*f-measure*
0	0	0	0.501532	0.859143	0.633344
0	5	0	0.559668	0.783036	0.652773
5	5	5	0.549815	0.798214	0.651129
15	25	5	0.615894	0.747321	0.675272
15	10	15	0.585333	0.783929	0.670229
15	25	15	0.695298	0.672791	0.683859
100	100	100	0.757858	0.569554	0.650350

Table 1: Precision and recall of the image retrieval

5.3.2 Semantic Hierarchy and Clusters

The results of the semantic hierarchy and cluster evaluation also depend on the parameters mentioned for image retrieval. The measurements listed in table 2 indicate that the best distinction between images showing the same objects and images showing different objects is achieved with low *minimal mcl cluster size*, that is, without outlier removal. The other parameters' values correlate with those of the image retrieval evaluation above.

mcl clustering threshold	*minimal mcl cluster size*	*minimal node size*	c_o	c_c	c_n	$c_o - c_n$
0	0	0	0.25075	0.25483	0.23430	0.01645
0	5	0	0.25707	0.26691	0.24857	0.00850
5	5	5	0.26129	0.27160	0.25347	0.00782
15	25	5	0.24118	0.24927	0.23345	0.00773
15	0	15	0.27757	0.28242	0.25897	0.01860
15	10	15	0.28285	0.29194	0.26884	0.01401
15	25	15	0.28571	0.29391	0.27563	0.01008
100	100	100	0.32578	0.34126	0.31711	0.00867

Table 2: Semantic quality measures

Varying the parameters most strongly influences the amount of the closeness measures, that is, all their values rise or drop somewhat consistently.

5.3.3 Visual Clustering

The 883 images, which our algorithm retrieves for the search term "food", and which users voted to contain food, were clustered into 21 visual clusters (see chapter 4.3). After 10 passes, we gain an average precision of 84.7% with a recall of 93.5%. The F-Measure is 88.9%.

Since, our visual clustering algorithm is supposed to be processed after the semantic clusters have already been formed, it is intended to be used on smaller sets of images. We perform a second measurement with 100 randomly picked images which are then clustered 100 times into 7 buckets. Here, we can achieve a precision of 87.9% while the recall falls to 82.2%. The F-Measure is 84.9%.

reduce to 3 pages, but not a must, we currently have 20 pages without abstract

6 Results Discussion

The results of our evaluation are not only affected by our algorithm, but also by the test set. The following sections give an inside to the reasons for unexpected and remarkable results.

probably write a little bit more

6.1 Test Set 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 definition 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”.
Another problem lies in the fact that pictures often contain small or processed items, which makes it hard to identify the exact contents of that picture. That is why, during test set creation participants could probably not see the described content or interpret the image different in contrast to the original tags of the image. 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 there are enough participants 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, e.g. *fall* as a verb for falling under the influence of gravity and as a noun for autumn. Second, words in other languages than English may be incorrectly matched if they exist in a different meaning in English, e.g. *gift* for present in English and poison in German. And third, the assumption that tags on the same picture are semantically close does not hold in all cases (like words with different meanings also known as homographs), e.g. a *cherry* on a plate made of *wood* would be assigned to the cherry tree meaning instead of cherry as a fruit.
The first two reasons can be addressed by using a more sophisticated ontology and similarity measure, but the third requires a rethinking of the Synset detection algorithm.

The evaluation also indicates the effectiveness of an additional semantic clustering via keyword clusters for the image retrieval. One assumption was, that if only few pictures are assigned to the same keyword cluster, they can be declared as outliers and therefore be deleted from the retrieval result. The parameter *minimal mcl cluster size* determines which clusters are deleted. The results show that increasing the *minimal mcl cluster size* parameter in fact also increases the F-Measure. The best results are achieved if the *minimal mcl cluster size* is greater than *minimal node size*. That means deleting pictures is more effective than merging with parent node.

6.3 Semantic Clusters

The closeness of nodes shows, that images, which were annotated as semantically similar, are in the tree closer to each other than as semantically not similar annotated ones. However, all values are rather close to average tree distance. Furthermore, the varying of parameters influences only the average tree distance, not the difference between distances. In our opinion, pictures, annotated as showing same object, should at least be in the same node, which means closeness should have value greater than 0.5. Images with same context should be in the same cluster, therefore, closeness should be 1. Unfortunately the results are best with other parameters than for a good image retrieval.

One problem, which we encountered, is the difficulty of defining "semantically similar". The second phase of our evaluation process included new participants, who did not know that all images contain food. That is why, some participants voted different kinds of food as same object. Others voted images as showing same object, only if they really showed the same object (both showing a strawberry). This is a problem of granularity, depending on how fine grained participants evaluated semantic similarity, the results differ. Participants later stated insecurity about these votes and own inconsistency. A solution could be a detailed description of semantic categories, but our aim was to achieve a quite intuitive clustering. For a significant evaluation, more participants are needed.

Our semantic evaluation method is a good method to get a quality measure of the complete semantic hierarchy. However, this makes it hard to make statements about the individual semantic clustering steps. For example, the results of the keyword based clustering highly depend on the quality of the calculated keyword clusters. Those are hard to judge in isolation, though.

6.4 Visual Clusters

The visual clustering shows good results in the evaluation, i.e. the clustering puts images a human considers visually different in the majority of cases into different visual clusters. Yet, in the case of 883 images, our algorithm separates the pictures in 23 different clusters, which may be a lot for humans to differentiate. Fortunately, the visual clustering is intended to run on smaller image sets, since the image sets are already semantically pre-clustered.

When performing the analysis on a smaller subset with 100 images, the precision went up to 87%. Additionally, the images were clustered in only 7 clusters. But, the recall fell to 82.2%, because less clusters lead to less differentiation.

one sentence why falling recall not bad, see last point in notes

7 Conclusion and Future Work

Within this work we described a new approach to cluster images in homogeneous groups by extracting semantic and visual information. Our goal was to assign every image a semantic meaning based on their meta information retrieved from the MIRFLICKR-1M file set and matching these to their meaning based on Synsets from knowledge base WordNet.
We built a tool which creates a search tree containing hyponyms and meronyms for a given search term and groups matching images into semantically and visually similar clusters.

The approach is a combination of algorithms which have not yet been combined in this way. Our aim was to retrieve training data for semantic image analysis. However, this work is only a starting point. Our tool permits a general differentiation of images. During our evaluation, we found out that, especially in the semantic analysis, challenges still exist concerning evaluation and level of granularity.

7.1 Semantic

To continue our work, there are several points of action. One main issue is the fact that our keyword clusters are of different levels of abstraction. Some clusters are too large, and could be separated, and others are too small, and should be merged. Another challenge is the correct synset detection. Additionally more meta information could be used for semantic clustering, like *groups* and *albums*. With the help of named-entity recognition, the *description* can be also valuable. As alternative to WordNet, it could be possible to use DBpedia[10]. Probably, even a combination can be helpful.

use more or other WordNet relations?

7.2 Visuals

Because of the challenges in semantic clustering, our visual clustering remains on a basic level. Therefore, further improvements are feasible in the extraction of visual features. The usage of high level features for visual clustering could lead to better results, but the risk exists that clustering results are no longer intuitive.
Another point of action is the calculation of the number of clusters for k-means. A basic adaptive k algorithm does not increment the quality of our algorithm, but a new abort criterion could be an improvement. Additional enhancements can be achieved by using a more sophisticated approach for late fusion, like ...

find the name of the late fusion algorithm, probably state why not yet used

[10]http://dbpedia.org/About

A Glossary

Late Fusion: combines single-feature clusters by intersecting them. Ensures that all images within a cluster are similar in both features and lead to less or equal to $n/2$ subclusters.

Synset: a particular concept which can be expressed by different terms but has one unique identifier. This identifier consists of the word most commonly used to describe the concept, the part of speech, and a number, e.g. drive.v.02.

WordNet: 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" (description on the official web page[11])

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

Leacock and Chodorow Similarity: finds shortest path between two concepts. Uses adapted weights and maximum path length as normalization factors. Is perceived as closer to human understanding than regular path ((BH01) and (PPM04))

Folksonomy: a collaboratively created content classification system derived from annotating and categorizing content with tags by users

Ontology: "an explicit, formal specification of a shared conceptualization" (G$^+$93)

Hyponym:

Meronym:

[11] http://wordnet.princeton.edu/

B Abbreviations and Acronyms

HSV	Hue, Saturation, Value
LCH	Leacock and Chodorow
MCL	Markov Cluster Algorithm
tf-idf	Term frequency and inverse document frequency

standardize references

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.

[dHIM05] Michiel de Hoon, Seiya Imoto, and Satoru Miyano. The c clustering library. *Inst. of Medical Sci., Human Genome Cent., Univ. of Tokyo, Tokyo*, 2005.

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

[G+93] Thomas R Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[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.

[JLM03] Jiwoon Jeon, Victor Lavrenko, and Raghavan Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126. ACM, 2003.

[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.

[PPM04] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[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.

[RSM94] Ray Richardson, Alan F Smeaton, and John Murphy. Using wordnet as a knowledge base for measuring semantic similarity between words. Technical report, Technical Report Working Paper CA-1294, School of Computer Applications, Dublin City University, 1994.

[SK04] Julian Sedding and Dimitar Kazakov. Wordnet-based text document clustering. In *Proceedings of the 3rd Workshop on RObust Methods in Analysis of Natural Language Data*, pages 104–113. Association for Computational Linguistics, 2004.

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

[WDS+01] Liu Wenyin, Susan Dumais, Yanfeng Sun, HongJiang Zhang, Mary Czerwinski, and Brent Field. Semi-automatic image annotation. In *Proc. of interact: conference on HCI*, pages 326–333, 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

fill index