Multimed Tools Appl (2018) 77:8847–8882
DOI 10.1007/s11042-017-4777-8

ConceptRank for search-based image annotation

Petra Budikova[1] · **Michal Batko**[1] · **Pavel Zezula**[1]

Received: 18 October 2016 / Revised: 10 March 2017 / Accepted: 28 April 2017 /
Published online: 16 May 2017

Abstract Multimedia information is becoming an ubiquitous part of our lives, which brings an equally ubiquitous need for efficient multimedia retrieval. One of the possible solutions to this problem is to attach text descriptions to multimedia data objects, thus allowing users to utilize traditional text search mechanisms. Search-based annotation techniques attempt to determine the descriptive keywords by analyzing the descriptions of similar, already annotated multimedia objects, which are detected by content-based retrieval techniques. One of the main challenges of this approach is the extraction of semantically connected keywords from the possibly noisy descriptions of similar objects. In this paper, we address this challenge by proposing the ConceptRank, a new keyword ranking algorithm that exploits semantic relationships between candidate keywords and utilizes the random walk mechanism to compute the probability of individual candidates. The effectiveness of the ConceptRank algorithm is evaluated in context of web image annotation. We present a complex image annotation system that includes the ConceptRank component, and compare it to other state-of-the–art annotation techniques.

Keywords Search-based image annotation · Content-based image retrieval · kNN classification · Biased random walk with restarts · Semantic analysis · ConceptRank

✉ Petra Budikova
budikova@fi.muni.cz

Michal Batko
batko@fi.muni.cz

Pavel Zezula
zezula@fi.muni.cz

[1] Masaryk University, Brno, Czech Republic

1 Introduction

Information is one of the most valuable assets of human society. Nonetheless, a piece of information is only useful if it can be found when it is needed. Modern technologies allow us to create and store enormous amounts of digital information, including complex formats such as multimedia. However, we are still struggling to develop efficient and user-friendly tools for effective management and retrieval of such data, which would allow us to fully exploit the accumulated information.

Although a lot of effort has been invested in the last decades into the development of similarity-based retrieval systems, it appears that keyword search remains the most natural way for people to access information. To be able to find a multimedia object by keyword search, the object needs to be described by sufficiently rich and precise text metadata. This is well known e.g. to photo-stock sites, which sell images that need to be located by keywords. As observed in [7], the average number of keywords used for description of web-stock photos is about 30.

When the text metadata is created manually, it is a time-consuming process that requires more effort than it may seem. To demonstrate this, we performed a small experiment, where university students were given a set of images and asked to provide the descriptive keywords. The participants were told that the keywords would be used for text retrieval, but we did not issue any recommendations on the number of keywords. As a result, the students usually provided only a few keywords per image, the average number being 5 (see Fig. 1). To come up with more keywords would apparently require more time and effort than they considered appropriate to invest.

In this situation, it is a logical step to attempt to create the descriptive keywords automatically – either entirely or, at least, in the form of keyword hinting with subsequent manual selection by the user. Indeed, the research in automatic multimedia classification has been ongoing for several decades, starting from dedicated (e.g. medical) classification tasks and expanding towards more general problems such as web image annotation. However, with the growing amounts of data to be processed and the increasing number of target classes from which we select (i.e. categories, labels, tags, etc.), new challenges appeared. In particular, the standard machine learning techniques seem to reach their limits with hundreds of target classes, while the required number of classes for web search can be thousands or even more [18, 49]. Therefore, alternative solutions need to be explored.

Fig. 1 Comparison of keyword annotations from different sources

A lot of attention has been recently focused on exploiting content-based retrieval techniques to extract information from vast amounts of user-provided data available on the web. This approach, usually denoted as *search-based annotation*, consists of two main steps: first, a set of objects similar to the annotation input is selected, using the content-based searching; second, keywords associated with the similar objects are submitted to some tag ranking function, which selects the most relevant ones for the annotation output. Although the search-based annotation methods are not yet sufficiently precise, the search-based paradigm has several desirable properties, in particular the scalability with respect to both data size and target vocabulary size. The principal challenge is now the selection of relevant keywords from the candidates provided by the content-based retrieval. Also, a number of additional issues need to be resolved, such as the selection of a suitable knowledge base, efficient data indexing, or the combination of various keyword relevance clues.

1.1 Our contributions

In this paper, we address the problem of automatic multimedia annotation on two levels: first, we propose a new *generic* semantic-mining algorithm for search-based annotation; second, we present a specific *image* annotation tool that provides descriptive keywords for web images. Subsequently, we should talk about *multimedia* annotation in the first part of the paper, and *image* annotation in the second part. However, we feel that this would add unnecessary confusions. Therefore, we will focus on image annotation throughout the paper, and we shall only emphasize the genericity of selected procedures in the text. The main contributions of the paper are:

- *ConceptRank algorithm:* During the annotation process, one of the crucial tasks is to compute a probability score for different candidate keywords that were associated with objects similar to the query. We introduce the ConceptRank algorithm, a generic procedure for computing the keyword probabilities that takes into account the content-based similarity of multimedia objects as well as various semantic relationships that exist between the candidate keywords. The relationships can be retrieved from an arbitrary semantic resource, e.g. the WordNet.
- *Description of a mature search-based annotation system:* To facilitate the search-based annotation, many issues need to be resolved, such as selection of the reference image database, efficient implementation of large-scale similarity retrieval, semantic ranking of candidate keywords, or selection of the final annotation output. In the second part of the paper, we present a mature system for image annotation, and discuss the design and implementation of individual components.
- *Extensive evaluation of the annotation system and its components:* To demonstrate the usefulness of the ConceptRank and to analyze its behavior in various situations, we performed extensive experiments on the ImageCLEF 2014 Scalable Concept Image Annotation dataset. At the same time, we assessed the effectiveness and efficiency of the whole annotation system. Our tool was compared to other participants of the ImageCLEF competition and also to the image labelling tool offered by the Google Vision API.

The paper is organized as follows. First, we formalize the problem of image annotation and provide a brief comparison of two orthogonal approaches – the model-based and the search-based paradigm. Next, we give a more detailed description of the search-based annotation principles, related work, and challenges. In Section 4, we introduce the ConceptRank algorithm and its utilization on semantic networks of related concepts. The following section

presents a real-world annotation system featuring the ConceptRank algorithm together with the WordNet lexical database. In Section 6, the performance of the whole annotation tool is evaluated and analyzed. In the conclusion, we outline a possible future cooperation between our annotation-based solution and state-of-the-art classifiers.

2 Preliminaries

Image annotation is an active field and various researchers may differ in their definitions of the annotation task. Therefore, let us begin by clarifying the specific problem that we want to solve. Next, we briefly survey two orthogonal approaches to the task and outline their strengths and weaknesses.

2.1 Problem formalization

In the context of this paper, the objective of image annotation is to provide a set of descriptive keywords that will characterize the visual content and semantics of some input image in the extent required by a given application. Although there already exist works that aim at describing images by coherent sentences, we feel that providing a rich and accurate set of keywords is still enough of a challenge. Keyword annotations are also more suitable for most applications.

The term *keyword* is used here to represent a single word or a short phrase acting as an atomic item for describing the image content. Due to the ambiguity of natural languages, the plain text representation may not be sufficient to determine the meaning of a given keyword – e.g. the keyword *bar* can refer to a drinking place, a piece of metal, music notation, or several other objects or activities. Therefore, the semantics of a keyword can by clarified by linking it to some external knowledge base (a Wikipedia page, some ontology class, etc.). If available, such links are very helpful during the annotation process.

Definition 1 A keyword is a single- or multiple-word label of image content. Its semantics can be specified by a link to some external knowledge base.

As stated above, the annotation process needs to take into account the specific application for which the annotation is formed. For instance, the query image in Fig. 2 can be described simply by the flower name, or by a set of keywords that specify its color, shape, and surroundings. All these descriptions are correct, but they are suitable for different situations. Therefore, we introduce the notion of a *target vocabulary* V, which is a set of keywords that are of interest for the given application. If it is not explicitly specified, we assume that the target vocabulary contains all English words. By the means of the target vocabulary, we can uniformly model annotation tasks with different scopes and domains, including traditional multi-label classification tasks such as the ILSVRC [39]. For web image annotation, explicit target vocabularies are considered e.g. in [11, 48].

Definition 2 A target vocabulary is a set of keywords that are eligible for annotation in a given application.

Now we can define the annotation task and its objective. In simple words, we are looking for a procedure that computes the relevance of individual vocabulary items with respect to a given query image.

Query image

A **Application**: keyword annotation for text search
Vocabulary: *all English words*
Relevant keywords: {flower, dandelion, plant, yellow, detail, ...}

B **Application**: plant identification
Vocabulary: *list of plant names*
Relevant keywords: {Taraxacum_officinale}

C **Application**: personal image labeling
Vocabulary: {animal, building, flower, person}
Relevant keywords: {flower}

Fig. 2 Image annotation in context of different applications

Definition 3 The annotation task is defined by a binary query image q and a target vocabulary V. The solution to the task is modeled by an *annotation function* $f_A : Image \times Keyword \rightarrow [0; 1]$, which for each keyword $c \in V$ computes the probability of the keyword being relevant for q.

Any keyword with a non-zero probability is deemed relevant for the image and may become part of the annotation output. However, only a subset of the relevant keywords is often presented to the user, especially in case of large vocabularies. This subset can be selected by limiting the number of the most probable keywords, defining a minimum probability threshold, or using some advanced mechanism that takes into account semantic relationships between words and their hierarchies.

We should also mention that in some situations, the query image may be already accompanied by some text information, either user-provided or automatically retrieved. The query image can also be accompanied by other types of metadata, such as the GPS location or EXIF. To cover those cases, the annotation input in the above-mentioned definition could be easily extended. However, since we do not work with additional input information in this paper, we prefer the simple definition of f_A.

2.2 Possible approaches

There are two fundamental approaches that try to transform visual image content into textual information, which are traditionally denoted as *model-based* and *search-based* annotation. The model-based paradigm makes use of various machine learning techniques, whereas the search-based approach exploits recent advances in content-based image retrieval.

The model-based annotation begins with a learning phase, when a correctly labeled training dataset is used by machine learning processes to create a statistical model for each concept from the target vocabulary. The models are then used during the actual annotation phase to decide the relevance of individual concepts with respect to a given query image. As surveyed e.g. in [58], numerous learning techniques have been studied in context of image annotation. Recently, very good results have been obtained by deep convolutional neural network classifiers [23, 44]. However, a key component for the model-based annotation is reliable training data, which is notoriously difficult to obtain. Furthermore, the learning phase is costly and any change of the target vocabulary requires re-training of the whole system. Also, with the growing size of the target vocabulary, it becomes difficult to train the classifiers both in terms of computation costs and class confusability [18, 49].

Search-based image annotation, also denoted as data-driven or model-free annotation, is an orthogonal approach to machine learning. It attempts to utilize the voluminous but

potentially erroneous information available in different web image collections and social networks. In time of the query execution, a content-based image search is initiated to search such resources for images that are visually similar to the picture being annotated, and the textual metadata of the resulting images is used to form the annotation [25, 54]. The underlying assumption is that a significant portion of visually similar photos should be also semantically related to the image that is being analyzed. The use of web data instead of dedicated training collections significantly lowers the barriers of building an annotation system. The search-based annotation also needs no learning phase and scales well to large vocabularies, since the vocabulary of web image databases that can be utilized to annotate images is potentially unlimited. However, the current precision and recall of search-based annotation methods is lower than the performance of state-of-the-art classifiers [22, 26, 60]. Therefore, the search-based annotations are currently more suitable for tag-recommendation tools than for a fully automated annotation service.

As we can observe in Table 1, the two above-described paradigms are in many aspects complementary. Therefore, we believe that the future of image annotation lies in combinations of the two approaches, as suggested e.g. in [3, 10, 17, 53, 59]. To make such schemes viable, it is necessary to continue developing both paradigms, especially the less efficient search-based techniques.

3 Search-based image annotation

In this paper, we focus on the development of search-based image annotation while keeping in mind the possible future fusion with dedicated classifiers for selected concepts. In this

Table 1 Comparison of the model-based and search-based annotation

	Model-based approach	Search-based approach
Principles	– use training data to create classifiers for vocabulary concepts (learning phase, offline) – run the classifiers to select relevant concepts for a given query	– employ similarity search over annotated data to find objects similar to a given query – mine the annotation from the descriptions of similar objects
Advantages	– mature technologies available – fast processing – high precision and recall	– reducing the reliance on cleanly labeled data, utilization of web data – no costly learning phase – scalability w.r.t. vocabulary size
Disadvantages	– requires reliable training data – extension of vocabulary requires costly re-training of classifiers	– slower annotation phase – lower precision and recall
Use cases	Annotation tasks with small domains, fixed vocabulary, and reliable training data – identification of people – classification of cancer cells	Annotation tasks with large domains and open/adaptable vocabulary – proposing keyword annotations for web image databases

section, we detail the principles of the search-based approach and discuss related work in this field.

3.1 Principles

The basic idea of the search-based paradigm is to annotate an unlabeled image by propagating the labels of community photos that are visually similar to the input image. Let $AIC = \{(i_1, desc_1), \ldots, (i_n, desc_n)\}$ be the reference collection of n images that are already associated with some type of textual data. This collection is typically pre-processed to enable efficient evaluation of similarity queries. The actual annotation process then consists of several phases, which are illustrated in Fig. 3. Let us now describe each of these phases in more detail and introduce some notation that will be used later:

- *Content-based image retrieval (CBIR)*: Using a suitable visual distance measure d_{vis}, the system first searches the annotated image collection for images similar to the query image q. The search returns a set of visually similar images together with their descriptions and their visual distances from q: $Sim_q = \{(i_{m_1}, desc_{m_1}, d_{m_1}), \ldots, (i_{m_k}, desc_{m_k}, d_{m_k}) | (i_{m_i}, desc_{m_i}) \in AIC, d_{m_i} = d_{vis}(q, i_{m_i})\}$
- *Candidate keyword mining and probability computation*: In the next step, the keywords that appeared in the annotations of images in Sim_q are collected. We shall refer to them as the *initial candidate keywords* Kw_q^{Init}. These candidate keywords are associated with initial probabilities, which are usually derived from some properties of Sim_q. Additional text mining techniques can then be applied to expand and refine the set of candidate keywords and to recompute the probabilities of individual candidates. At the end of the second phase, we obtain Kw_q^{Final} – a final list of keywords and their probabilities.
- *Mapping to the target vocabulary*: If the target vocabulary V differs from the vocabulary of images in AIC, the candidate keywords retrieved by the previous step may not be eligible for the final annotation. In such case, the keywords from Kw_q^{Final} have to be mapped to appropriate keywords from V, and the probability of the target keywords needs to be determined. At the end of this phase, the value of $f_A(q, kw)$ is available for all keywords from the target vocabulary.

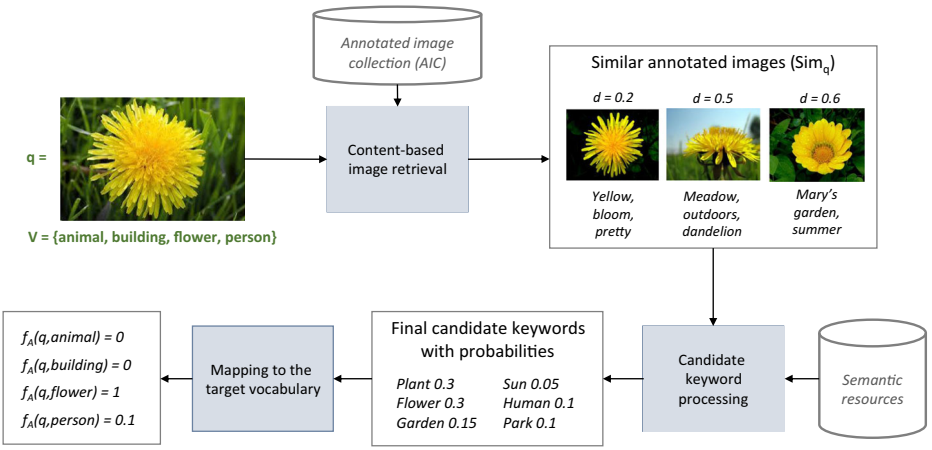


Fig. 3 General scheme of the search-based annotation

– *Selection of output*: If the set of relevant target keywords is reasonably small, it can be directly returned to the user/application. Otherwise, it may be necessary to select a subset of relevant keywords as the final annotation.

3.2 Challenges and related work

To make the above-described scheme work, a number of issues has to be resolved. In the following, we introduce the main challenges and discuss existing work in individual areas. The related work for the computation of f_A is especially relevant for Section 4 of this paper, where we introduce a new algorithm for candidate tag probability computation. The other challenges will be met again in Section 5, where we present a complete annotation system.

3.2.1 Data acquisition

The reference dataset (AIC) used for content-based retrieval significantly influences the quality of annotations. One of the important factors is the size of the dataset and its scope – the broader is the domain D to be covered, the more images are needed to provide examples for different visual forms of the semantic concepts from D. At the same time, the quality of both the visual and text information contained in the AIC should be high to allow meaningful visual comparisons and text analysis.

Existing solutions to reference data acquisition can be divided into three groups. In the first group we put solutions that employ web images with the surrounding texts dumped from web search engines [12, 54]. Such images can be obtained in large quantities, but the quality of both image and text is unreliable and there is the additional task of mining keywords from the web page. The second group contains collections retrieved from various web image galleries, e.g. Flickr [1, 22, 51, 60]. This approach is popular due to the fact that the images are already tagged. Finally, there are also collections that were created for the specific purpose of supporting image annotation and classification. Two significant examples both focus on linking images with the WordNet lexical database: the authors of [47] automatically collected 80 million images representing all non-abstract WordNet noun classes, whereas the ongoing ImageNet project with crowdsourced quality control currently reports over 14 million images linked to more than 20000 WordNet classes [13].

3.2.2 Effective and efficient CBIR

In the CBIR phase, we need to find images that are visually—and hopefully also semantically—similar to the query image. This requires suitable visual descriptors and distance function, which are used to evaluate the similarity. The similarity search needs also to be efficient, since in most cases the users expect real-time interaction with the annotation system. Therefore, we need effective and efficient data indexing and retrieval methods that are able to process large quantities of images.

Basically, there are two types of visual image descriptors: local descriptors identify and characterize important points in the image, whereas global descriptors provide aggregated information about the whole image. A general survey of various descriptors can be found in [14], whereas [58] focuses on descriptors used in image annotation. In both areas, a distinct success has been achieved lately by descriptors produced as a by-product of deep neural network classifiers, in particular the DeCAF global descriptors [15].

The efficiency of similarity searching is determined by the indexing technique used for data organization [57]. Recent advances in the area of multimedia indexing include

methods based on vector space partitioning [35], permutation-based metric space indexing [36, 37], or similarity-based hashing [29]. Alternatively, the bag-of-words approach utilizes text-search techniques on descriptors converted to visual words [41].

It is also interesting to consider the type of similarity query employed for the identification of relevant images. A vast majority of annotation techniques utilize the *k-nearest neighbor query*, which returns the k most similar images. As an alternative, [27] suggested to use *range queries* and claimed that the optimal distance threshold can be determined more precisely that the optimal value of k.

3.2.3 Effective and efficient identification of relevant keywords

The search-based annotation is based on the assumption that a significant portion of initial candidate keywords is relevant for the query image. However, there are two factors that introduce noisy and irrelevant initial keywords: the uncertain quality of image descriptions in AIC, and the semantic gap that may cause CBIR to return semantically irrelevant images [42]. Therefore, a refinement stage is necessary to identify the keywords which are strongly correlated and reject others. At the same time, the initial candidate keywords can be expanded by related words from other sources. Finally, the probability of relevance needs to be estimated for all candidate keywords, which are then mapped on the target vocabulary. There is thus a demand for robust models that are able to combine different relevance clues to rank candidate annotations. The models also should be efficient to allow real-time responses of the whole annotation system.

The first task of the text processing phase is to select the initial candidate keywords. This is trivial if the images in AIC are annotated by keywords; otherwise, meaningful n-grams need to be extracted from the image descriptions [12, 54]. Next, the set of candidate keywords can be enriched by related concepts from various resources. [1] proposed to consider WordNet synonyms of the initial keywords, [32] and [21] utilized words frequently co-occurring in AIC with the initial candidates, and [60] selected additional keywords from the results of a web search that used the initial candidate keywords as the query.

A baseline strategy for the estimation of initial candidate keyword probabilities is to consider the frequency of each keyword within Sim_q. Individual keyword occurrences are often weighted by the visual distance or rank of the respective image [34, 51]. Another frequent measure of keyword relevance is the observed co-occurrence of a given keyword with other candidate keywords [28]. Several recent papers proposed to utilize topic modeling and aggregate keywords with respect to common topics learned from web semantic resources [12, 54]. Many solutions model the relationships between keywords (and possibly also images) by weighted graphs and utilize different graph algorithms to determine the probability of individual nodes. In particular, [55] performed cluster analysis and used only the biggest clusters for tag transfer, whereas [30, 52, 60] employed various adaptations of the Random Walk with Restart. Yet another approach was used by [56] who proposed to measure the relevance of each candidate keyword by running a text-based image search and comparing the retrieved images to the query.

3.2.4 Selection of output annotation

As discussed earlier, unlimited target vocabularies need to be considered in some use-cases, e.g. the web image annotation. In case of general-purpose image description, there may literally be a thousand words that are relevant to its content. However, users typically do not want to scan hundreds of keywords. This introduces a new challenge that was unknown to

the traditional classification problems – the task of selecting the keywords that are not only relevant for the image but also relevant to the user.

The question of output keyword selection has not yet been much discussed in the literature. Most works either do not consider this problem at all, or return a fixed amount of the most probable keywords [33, 54] and [20] proposed to use probability thresholds instead of the fixed size limit for the selection of output keyword. On the semantic level, [22] suggested to prefer more specific keywords, determined by the WordNet hierarchy.

4 ConceptRank

In this section, we introduce a new method for computing the probability scores of candidate keywords provided by similar images. Our algorithm takes into account semantic relationships between the candidate keywords and attempts to simulate a human reasoning process by a random walk over a graph model of the keywords and their relationships. Since our algorithm was inspired by the famous PageRank algorithm [6] and deals with semantic concepts, we denote it as ConceptRank. In the following, we first provide an intuitive overview of the approach, then we describe in detail our model of keyword relationships and the algorithm for computation of keyword probabilities.

4.1 Overview

In the text processing phase of the search-based annotation, the computer is basically given the following task: *given the set of initial keywords, try to guess what is in the image that is similar to images described by these words*. If the same task was given to a person, he or she would start to mentally connect the keywords by semantic relationships and look for repeating themes. For instance, when we consider the initial keywords from Fig. 3 we immediately realize that "a dandelion is a flower", "there are flowers in a garden" etc., and the themes *garden* or *plant* quickly come to mind.

In the proposed solution, we attempt to simulate the above-described human reasoning in the computer. Although the machine lacks the knowledge about real-world relationships that people obtain by lifelong learning and experience, it can exploit various semantic resources that have been developed for language processing, semantic web, or AI purposes. There are numerous language models and ontologies that organize knowledge in semantic hierarchies that are natural to human cognition, and it has been demonstrated that utilization of such hierarchies can improve the quality of automatic image understanding [48]. Using the semantic knowledge sources, the computer can look for connections between the candidate keywords and identify additional candidates in a way that is very similar to the human thinking process. To estimate the probability of individual candidate keywords, we can then utilize the initial probabilities determined in previous annotation phases as well as the relationships between keywords. For the actual computation we employ the Random walk algorithm that has been used with great success in similar tasks.

The ConceptRank keyword processing can follow directly after the CBIR phase, or there can be any number of keyword processing steps in between. In any case, from the previous annotation step the ConceptRank receives a set of candidate keywords and their associated scores that represent the current estimates of individual keywords' probability: $Kw_q^{CR-Input} = \{(kw_1, score_1), \ldots, (kw_n, score_n)\}$. These will be analyzed using a semantic resource S; for simplicity, we only consider one semantic resource in our

discussion although the model can be easily adjusted for more knowledge sources. The actual analysis then consists of two phases. First, the input keywords retrieved from $Kw_q^{CR-Input}$ are mapped to semantic concepts from S, and a semantic network of interrelated concepts is constructed. Afterwards, an adapted Random walk algorithm is employed to determine the probabilities of individual concepts in the network. Both phases are described in detail in the following sections.

4.1.1 Innovations of the ConceptRank technique

Although various adaptations of similarity graphs and the Random walk algorithm have been used in former works for the modeling of keyword relationships and probability computations [30, 52, 60], the ConceptRank technique is innovative in several aspects. Most importantly, we apply the random walk algorithm on top of a graph model of semantic relationships between the candidate keywords, which to the best of our knowledge has not been studied before. Semantic hierarchies were considered for the modeling of keyword relationships in other contexts [40] as well as in several previous works on search-based image annotation [22, 28], but only to enrich the set of candidate keywords or to compute some static scores of individual candidates. However, since the semantic relationships between candidate keywords may be rather complex, we believe that the computation of probabilities needs to be incremental and allow mutual influencing between keywords. Let us consider a simple example of two candidate words *flower* and *tulip*: we cannot determine the probability of *flower* unless we know the probability of *tulip* and vice versa, since these concepts are interrelated and any update of probability of one concept should reflect on the probability of the other. Therefore, in the ConceptRank technique we include the relationships directly into the semantic model of candidate keywords, and our random-walk-based algorithm allows mutual influencing of related concepts during the keyword probability computation.

We also propose a novel way of integrating the relevance clues from previous annotation phases into the Random walk computation. The main motivation of this step is to maximize the synergy between the content-based (visual) and semantic information during the annotation processing. If we did not include the initial keyword relevance scores into the ConceptRank computation, the previous annotation phases would only filter out the less probable candidates, but the final keyword ranking would be purely semantic. In our approach, however, all information about candidate keywords that was accumulated during various annotation phases is exploited and merged during the final keyword probability computation. The presented solution is generic and can work with any number and type of word-to-word relationships.

4.2 Semantic network

The ConceptRank semantic network is a data structure that is used for recording all available information about candidate keywords and their relationships. Specifically, we utilize a directed multigraph where weights can be associated with both nodes and edges. Nodes represent candidate semantic concepts, edges are formed by the semantic relationships. The weight of a node represents the current estimate of the semantic concept probability, whereas the weight of edge $U \rightarrow V$ expresses the conditional probability of concept V being relevant given that U is relevant ($P(V|U)$). In Fig. 4, a small example of a semantic network is provided, depicting several semantic concepts encountered during the annotation of the dandelion image. Concepts *dandelion* and *flower* are connected by edge *hasInstance*

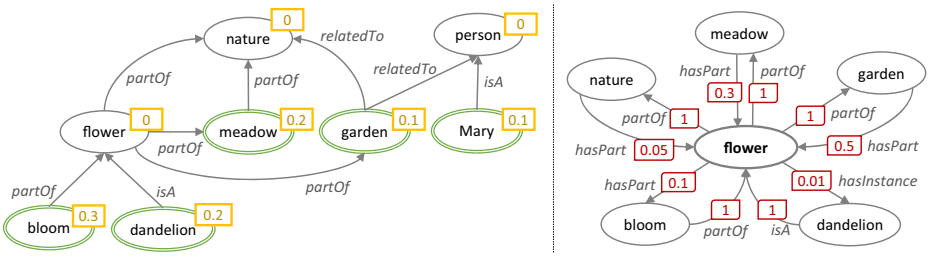


Fig. 4 Example of a semantic network: in the *left*, several nodes are shown together with node weights and selected edges; initial nodes are highlighted by the *green* (*double*) border; in the *right*, a detail of a selected node is provided including edge weights

with weight 0.01, since a dandelion is only one of many possible flower types;[1] however, each dandelion is a flower, therefore the reverse *isA* edge carries weight 1.

4.2.1 Initial concept nodes

In the beginning of the text processing phase, it is necessary to map keywords from $Kw_q^{CR-Input}$ to semantic objects from S, thus creating the initial concept nodes that will be used as seeds for the semantic network. Each initial node is associated with a non-negative weight that represents the probability of the respective keyword determined by previous annotation phases. If multiple semantic concepts are associated with a single keyword, which often happens due to language ambiguity, the original keyword probability needs to be distributed among all concept nodes that are created for this keyword.

4.2.2 Construction of the network

Using the initial concepts as starting points, we want to construct a network of semantically connected concepts by exploiting the information from S. However, there may be many types of links in S, and some of them may not be relevant for the annotation. Therefore, let $Rels$ be a set of relationship types from S that have been selected for the network construction.

Each of the relationships from $Rels$ can be exploited for two purposes: 1) discovering links between existing candidate concepts, and 2) identification of new candidates. However, if all relationships were used for both purposes, the semantic network might soon contain all concepts from S, which is usually not desirable. Therefore, we divide the relationships into two groups: *expansion relationships* are used for both relationship discovery and additional candidate selection, whereas *enrichment relationships* only add new links between existing network nodes. A typical example of an expansion relationships is concept generalization; if there is a concept node *flower* we want to add *nature* to the network because it is likely that other candidate concepts will link to it. However, there is no point in adding concept nodes for all flower subspecies, therefore specialization should be used in the enrichment mode.

[1]The value 0.01 associated with the *hasInstance* edge has no real justification here, its only purpose is to show that some connections are much less reliable than others. The exact edge weighting mechanism will be discussed later.

Algorithm 1 Semantic network construction

```
Input  : initObjectsWithProb – set of initial objects with probabilities,
         S – semantic resource,
         rels – set of selected relationship types
Output: semanticNetwork – the semantic network
begin
    queue = initObjectsWithProb.getObjects();
    while (not queue.isEmpty()) do
        o = queue.pop();
        semanticNetwork.addNode(o);
        foreach (r in rels) do
            foreach (o2 in S.getConnectedObjects(o,r)) do
                if (semanticNetwork.contains(o2)) then
                    semanticNetwork.addEdge(o,o2,r,computeWeight(r,...));
                else
                    if (r.isExpandingRel) then
                        queue.add(o2);
                        semanticNetwork.addNode(o2);
                        semanticNetwork.addEdge(o,o2,r,computeWeight(r,...));
                    end
                end
            end
        end
    end
end
```

The construction of the semantic network is formalized in Algorithm 1. In the beginning, we create a network node for each initial concept and put all these nodes into a queue to be processed. Then we successively remove and process nodes from the queue: for each node n and each relationship type $r \in Rels$, we check outgoing r-links from n and according to the type of r, add relationships and eventually new nodes to the network. Any new nodes are also enqueued for further processing. For each new node and relationship, it is necessary to determine their weight. In case of new nodes, the weight is always 0, since we have no clues for the node relevance during the network building phase. In case of relationships, we estimate the conditional probability with respect to the relationship type. At the moment, we support two weighting schemes – 1) a constant weight (e.g. the *isA* relationship always receives weight 1), and 2) a constant divided by the number of related nodes (used e.g. for *hasInstance* relationships; intuitively, $P(dandelion|flower)$ should be proportional to the number of flower species).

4.3 ConceptRank algorithm

When the semantic network is created, we obtain a rich set of candidate semantic concepts linked by relationships. For some of the concepts we have probability estimates from previous annotation phases, other concepts have zero starting probability. Now, we would like to update the probabilities of all nodes, taking into account the initial probabilities and the semantic links which transmit the scores between nodes. Since the network nodes mutually influence each other's probability, we need to find a steady state of this system.

For this purpose, we have chosen the random walk with restarts (RWR), an algorithm that was successfully used in many similar scenarios including the famous PageRank [6, 24]. As discussed in [46], the relevance score defined by RWR has many good properties: compared with pair-wise metrics, it can capture the global structure of the graph; compared

with traditional graph distances (such as shortest path, maximum flow, etc.), it can capture multi-facet relationships between two nodes.

4.3.1 Random walk with restarts

There are several algorithms that apply the random walk idea in various application areas, which differ in some details. Here we focus on the PageRank version, which inspired our research. PageRank evaluates the importance of nodes in a "web" graph, where nodes represent individual web pages and edges are defined by hyperlinks. The basic idea is to compute the probability that a user, who is browsing the web, reaches any given page. The virtual user starts in an arbitrary node and moves through the graph; whenever there are k possible ways, he chooses one of them with probability $1/k$. The random walk score of each node is defined as the steady-state probability that the user ends up in the particular node.

Let M be the transition matrix of a web graph G defined as follows: M is an $n \times n$ matrix, where n is the number of nodes in G. The element m_{ij} has value $1/k$ if node U_j has k outgoing edges and one of them is to node U_i; otherwise, $m_{ij} = 0$. Then the desired steady-state vector r of node scores should satisfy the following equation:

$$\mathbf{r} = M.\mathbf{r} \tag{1}$$

The iteration of (1) can be guaranteed to converge if the matrix M is column-stochastic and primitive, which corresponds to a strongly connected graph G where all nodes have some outgoing edges and the sum of outgoing edge weights for each node is 1. The desired vector $\mathbf{r}$ is then the principal eigenvector of M and it can be computed by repeatedly multiplying a random initial vector by M until the steady state is reached. However, many web graphs do not satisfy the necessary conditions; for instance, in Fig. 5a the node C is a so-called "dead-end" – it is referenced by multiple pages and should receive high score but,

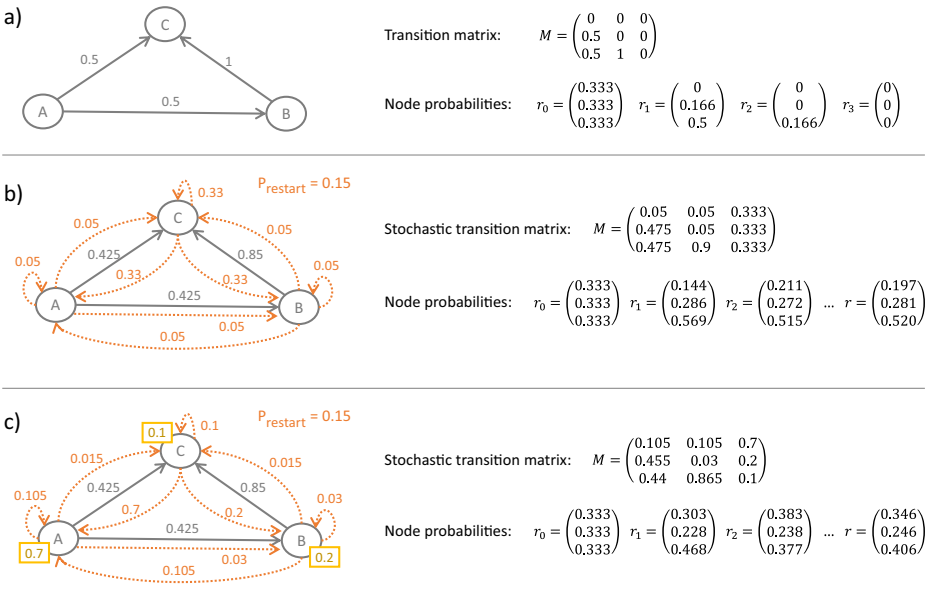


Fig. 5 Three types of random walk: random walk without restart (**a**); random walk with restart (**b**) – the restart edges are depicted by *orange* (*dotted*) line; random walk with biased restart (**c**) – the restart probabilities reflect the node weights of the target nodes

due to no outgoing edges, the score "leaks" out of the model. To overcome such problems, the PageRank authors further allow the surfing user to decide in each step whether to follow the links, or to randomly restart in any node. This can be understood as introducing additional edges into the graph G. With the possibility of restart, the simulation of a real user behavior is more realistic and the desired properties of the graph are achieved (Fig. 5b). The variable $P_{restart}$ determines the relative importance of the original and restart edges.

Let $\mathbf{u}$ be a vector of length n with all values equal to $1/n$. This represents the random restart edges, which are the same for all nodes. Then the new transition matrix M' is defined as follows: if there are no outgoing edges from node j, $M'_{ij} = \mathbf{u}_i$; otherwise, $M'_{ij} = (1 - P_{restart}).M_{ij} + P_{restart}.\mathbf{u}_i$. The following equation then can be guaranteed to converge for any graph:

$$\mathbf{r} = M'.\mathbf{r} \tag{2}$$

The original PageRank algorithm works only with the graph structure and does not assume any edge weights in G. However, it can be directly applied also to weighted graphs where the sum of weights of all outgoing edges for each non-leaf node is 1.

4.3.2 Biased random walk

In the basic PageRank, it is assumed that no prior knowledge about the importance of individual pages is available, and thus all nodes enter the computation as equal. However, there are also advanced versions of the algorithm that allow to prioritize some of the nodes. This can be done by biasing the restart vector $\mathbf{u}$. For instance, in the TrustRank algorithm the probability of restarting is no longer the same for all nodes – some trustworthy pages are more likely to be targeted by the restart [19]. In general, a biased restart vector $\mathbf{u}$ can contain any probability distribution such that $\mathbf{u}_i \geq 0$ and $\sum_i \mathbf{u}_i = 1$.

4.3.3 ConceptRank

In the ConceptRank algorithm, we utilize the biased random walk with restarts to compute the probability of nodes in the semantic network. Instead of the random user browsing the web, we model a random association that explores the candidate concept space. We assume that the probability of the association jumping from node U to node V is proportional to the weight of the edge $U \rightarrow V$. Let us remember that the weight of $U \rightarrow V$ was defined as $P(V|U)$, i.e. the conditional probability that concept V is relevant given that U is relevant. When the random association visits node U, it is natural that nodes with high $P(V|U)$ should be preferred in the next step of the simulated thinking process.

Apart from edge weights, the semantic network also contains node weights that carry important information from previous annotation phases and should be considered during the random walk computation of node probabilities. To allow this, we bias the restart vector so that it reflects the initial node weights, as illustrated in Fig. 5c. The restarts are more likely to target nodes with high initial weights, which subsequently increases the score of these nodes in the random walk computation.

The main steps of the ConceptRank procedure are summarized in Algorithm 2. First, the initial node weights are normalized so that their sum is 1, and used to determine the biased restart vector. Next, the transition matrix is constructed from the semantic network. The influence of individual relationship types can be adjusted by weight parameters. Finally, the transition matrix is combined with the biased restart vector so that the resulting matrix is

column-stochastic. Noticeably, after the last step the final matrix has the same properties as the PageRank matrix constructed from the web graph, so the random walk computation of the node probabilities can be performed in exactly the same way.

The probability of restart is determined by the parameter $P_{restart}$, which moderates the influence of the initial node weights. The eigenvector of the resulting matrix then contains the new relevance scores for all concept nodes in the semantic network. In the end, the ConceptRank returns a set of pairs $(s, P(s))$, where $s \in S$ is a semantic concept and $P(s)$ is the probability score for s.

Algorithm 2 ConceptRank computation

Input : semanticNetwork – *the semantic network*,
rels – *set of selected relationships and their weights*,
restartP – *probability of the random restart*
Output: semanticObjectsWithProbs – *set of candidate semantic concepts and their probabilities*

```
1  begin
2      numberOfNodes = semanticNetwork.getNodes().size();
       // construct the restart vector
3      restartVector = new Vector(numberOfNodes);
4      for (i=0; i ≤ numberOfNodes; i++) do
5          restartVector[i] = semanticNetwork.getNode(i).getInitialProbability();
6      end
7      restartVector.normalize();
       // construct the transition matrix
8      transitionMatrix = new Matrix(numberOfNodes,numberOfNodes);
9      foreach (r in rels.getRelationshipTypes()) do
10         relMatrix = constructSingleRelationshipMatrix(semanticNetwork,r);
11         transitionMatrix.add(relMatrix*rels.getWeight(r));
12     end
13     transitionMatrix.columnNormalize();
       // combine the restart and transition edges so that the resulting
          matrix is column-stochastic
14     finalMatrix = new Matrix(numberOfNodes,numberOfNodes);
15     for (i=0; i ≤ numberOfNodes; i++) do
16         if (transitionMatrix.getColumn(i).getColumnSum() == 0) then
17             finalMatrix.setColumn(i, restartVector.transpose());
18         else
19             finalMatrix.setColumn(i,
20                 (1-restartP)*transitionMatrix.getColumn(i) + restartP*restartVector.transpose());
21         end
22     end
       // compute the principal eigenvector and construct the result
23     nodeProbs = finalMatrix.getPrincipalEigenvector();
24     for (i=0; i ≤ numberOfNodes; i++) do
25         semanticObjectsWithProbs.add(semanticNetwork.getNode(i), nodeProbs[i]);
26     end
27 end
```

4.4 Efficiency issues

Image annotation is often desired in interactive applications, therefore we need the keyword processing to be effective as well as efficient. If applied as described above, both the semantic network construction and ConceptRank computation can be computationally expensive. In case of network construction, the complexity grows with the number of initial nodes and the number of relationships to be explored. The ConceptRank computation costs

are determined by the eigenvector decomposition, which is cubic to the size of the semantic network.

Therefore, for real-time applications we propose two approximations of the ConceptRank technique. First, we introduce an upper limit on the number of initial nodes of the semantic network. If more initial nodes are available, only the given number of the most probable ones enter the network building phase. Second, we approximate the eigenvector decomposition by iterative matrix-vector multiplication that is repeated until the vector is close to unchanged at one iteration. Both our experience and the PageRank studies (e.g. [24]) show that 50–70 iterations are sufficient to achieve a very good approximation of the exact eigenvector probability distribution.

5 Application: web image annotation with WordNet ConceptRank

The ConceptRank algorithm is implemented as a part of the MUFIN Annotation Framework [3], which is a modular system designed to support different types of annotation tasks. In this section, we present a complex annotation tool built in this framework, which performs online web image annotation using state-of-the-art content-based retrieval and the ConceptRank algorithm.

5.1 MUFIN Annotation Framework

As we observed in the introduction, multimedia information is becoming ubiquitous and automated annotation tools are desired in many situations. Even though the annotation task details may differ for individual use-cases, the basic structure of the software solutions is usually very similar. Following this observation, we have designed and implemented the MUFIN Annotation Framework [2, 3], which supports a wide range of annotation tasks by defining a modular and flexible architecture where individual components can be easily combined and reused. Several search-based algorithms and candidate keywords processing components are currently available within the MUFIN Annotation Framework, as well as a pipelining mechanism that passes a central *annotation record* object between the components.

5.1.1 ConceptRank component

The ConceptRank component of the MUFIN Annotation Framework encapsulates the semantic analysis described in Section 4. It accepts an annotation record containing a set of candidate keywords and returns an updated record with a new candidate set that contains semantic concepts. The component provides a generic support for creating the semantic network, and the actual algorithm for node probability computation. To utilize the ConceptRank component with any particular semantic resource S, it is only necessary to provide the subroutines specific to S, in particular the transformation of keywords to semantic objects and the retrieval of semantic relationships.

5.2 MUFIN web image annotation tool

As discussed earlier, the search-based annotation is mainly suitable for tasks with large vocabularies. Our prototype application is therefore a keyword hinting tool for web images, which suggests keyword annotations that can be exploited by text search to access the image.

Table 2 Overview of the technologies and resources used in MUFIN Image Annotation

Annotation tool component	Existing solutions	Our solution
Reference image collection	Flickr, ImageNet, random web images	**Profiset**
Visual descriptors	Local descriptors (SIFT, SURF, ...), global descriptors (DeCAF, MPEG7, ...)	DeCAF
Data indexing method	Vector-space partitioning, metric space partitioning, hashing, bag-of-words approach	PPP-codes (metric space)
Type of similarity query	kNN-query, range query	kNN-query
Semantic resources	Text co-occurrence statistics, WordNet	WordNet, **VCO**
Keyword relevance assessment	Simple (frequency/distance-based), semantic (co-occurrences, topic modeling, graph analysis)	**ConceptRank** (semantic graph)
Final answer selection	Fixed-size, threshold-based	Fixed-size

The middle column recapitulates the most significant representants of existing solutions; for more details and references, please refer to the related work survey in Section 3.2. Original components proposed for the MUFIN Image Annotation are highlighted by bold face

Such tool can be used e.g. by contributors of image-stock sites, but also for personal photo tagging. The keywords should serve for general text search, so there is essentially no limitation of the target vocabulary. However, we also want to evaluate the effectiveness of our annotation tool, which is quite difficult with unlimited vocabularies. Therefore, we include the option of mapping the annotation to a restricted vocabulary.

The basic structure of the MUFIN Image Annotation software reflects the general architecture of any search-based annotation system, as depicted in Fig. 3. Let us now focus on the specific technologies and data sources that are exploited in individual annotation phases. Table 2 provides a brief summary of available techniques and our choices, the following paragraphs discuss our decisions in more detail.

5.2.1 Data acquisition & content-based image retrieval

Before we can start searching for similar images, it is necessary to choose a suitable dataset of annotated images. The MUFIN Image Annotation tool currently utilizes the Profiset, a collection of 20 million photos with rich keyword annotations that were downloaded from the Profimedia image-stock site[2] and are available for research purposes [7]. The image-stock data represent a very good trade-off between data quality and quantity; although it is collected in an unsupervised way, the authors were financially motivated to provide high-quality pictures and annotations. The Profiset images cover a wide range of topics, including people, nature, buildings, or objects, which makes the collection a suitable resource for general-purpose image annotation. We have also experimented with other annotated image collections, namely the ImageNet and Flickr, but in both cases the quality of annotations was noticeably lower. The ImageNet database provides a single label per image, which is too scarce for search-based annotation, whereas the Flickr descriptions are too erroneous.

To evaluate the visual similarity of images, we employ the cutting-edge DeCAF descriptors that were extracted from the whole Profiset collection [38]. To compare the extracted vectors, we utilize the Euclidean distance as recommended by [15]. The dataset is indexed

[2] http://www.profimedia.com

by the PPP-Codes technique [37], which allows efficient evaluation of kNN queries that are used to retrieve the set of visually similar images Sim_q.

5.2.2 Candidate keyword processing

From the content-based retrieval, we obtain a set of described images and their respective distances from the query: $Sim_q = \{(i_{m_1}, desc_{m_1}, d_{m_1}), \ldots, (i_{m_k}, desc_{m_k}, d_{m_k}) | (i_{m_i}, desc_{m_i}) \in AIC, d_{m_i} = d_{vis}(q, i_{m_i})\}$. The set of initial candidate keywords Kw_q^{Init} is formed strightforwardly by merging all keywords found in Sim_q. To determine the initial probabilities of keywords, two alternative methods were implemented for experimental comparison. In the first, we utilize plain keyword frequencies; the second takes into consideration also the visual distances, prioritizing keywords belonging to more similar images.

The initial keywords are forwarded to the ConceptRank component of the MUFIN Annotation Framework. For the purpose of general image annotation, we have decided to utilize the WordNet lexical database as the source of semantic information.

5.2.3 WordNet

WordNet is a comprehensive semantic tool interlinking a dictionary, thesaurus, and a language grammar book [16]. It organizes individual words into synonym sets called *synsets*; each synset represents one underlying lexicalized concept. On top of synsets, several types of conceptual relationships are endoded, such as hypernymy, meronymy, or antonymy. In the current version, WordNet contains more than 150,000 English nouns, verbs, adjectives, and adverbs.

WordNet is a valuable resource for the annotations for several reasons. First, it is a manually created, rich and precise database of general English words. Second, it contains several relationships that are useful for image content analysis. And third, WordNet is linked to numerous similar structures for other languages as well as ontologies such as YAGO [43], which can be easily integrated in the future as additional resources for the annotations.

5.2.4 ConceptRank with WordNet

For the ConceptRank analysis, it is first necessary to transform the initial candidate keywords into candidate synsets. Due to natural language ambiguity, there are often more possible meanings of a specific word, and consequently more possible synsets. During the transformation to synsets, we do not attempt to decide which of the meanings is the correct one; this will be done later during the ConceptRank score computation. Instead, we allow that each keyword may be represented by several candidate synsets. The original keyword's score is distributed among the related synsets with respect to the synset frequencies, which are also available in WordNet.

The initial synsets become the base nodes of a semantic network. To build the network, we can exploit various WordNet relationships. Since the image annotations are mostly composed of nouns, we currently limit our attention to noun-related relationships. Among these, the most interesting ones are hypernymy (generalization) and hyponymy (specialization), which form the IS-A hierarchy of all nouns. Furthermore, we also employ meronymy (part-to-whole) and holonymy (whole-to-part). Hyponymy and holonymy are utilized in the enrichment mode, e.g. only add edges between existing network nodes, whereas hypernymy and meronymy may add new nodes into the network as well as edges. The relative

importance of individual relationships is determined by parameters, which will be analyzed in Section 6.

5.2.5 Visual concept ontology

Although the WordNet hypernymy/hyponymy hierarchy provides us with valuable information, some of its properties are not optimal for the ConceptRank. In particular, all branches ultimately go up to a single root node *entity*. When exploiting the hypernymy relationship, this would make all network nodes semantically interconnected, which is not desirable. Therefore, we need to limit the hypernymy exploration. Since the level of detail in individual branches of the WordNet hierarchy significantly differs, it is not possible to simply select a cut-off level.

To address this issue, we have created the Visual Concept Ontology (VCO), which provides a high-level categorization of WordNet noun synsets [5]. Apart from defining a set of semantically consistent top-level categories, it also provides some new conntections between synsets that are missing in the WordNet. The VCO is used together with WordNet in the semantic network building phase.

5.2.6 Mapping to target vocabulary

The ConceptRank component returns a set of candidate synsets with associated probability values. In the next step, we need to transform the synsets back to keywords, which are expected as the annotation output. As discussed earlier, the annotation task may be defined with or without an explicit target vocabulary, and the MUFIN Image Annotation tool supports both modes. If the target vocabulary is defined, the candidate synsets have to be matched to it. Otherwise, we need to select representative keywords for the candidate synsets from the entire English vocabulary.

When the target vocabulary is defined, the synset-to-keyword transformation can be done straightforwardly, using the WordNet mapping. Whenever a candidate synset is associated with a vocabulary keyword, the score of the keyword is increased by the score of the synset.

If the target vocabulary is not defined, the selection of output keywords becomes more difficult. First, it is problematic to decide how many keywords should be chosen for a given synset, and which ones. Also, the semantic network construction introduces many candidate synsets that are important for the computation of probabilities, but not for the output – for instance, there are 6 intermediate synsets in the WordNet hypernymy hierarchy between *dog* and *animal*, including concepts such as *placental mammal* which are not interesting for the user but cannot be easily detected. These issues represent a challenge for future development of the MUFIN Image Annotation tool; at the moment, we avoid them by a simple approximation – we create an implicit target vocabulary that is composed of the initial candidate keywords Kw_q^{Init}. Obviously, with the implicit vocabulary the ConceptRank procedure cannot introduce new keywords to the final result; however, it is still highly useful for identifying the most relevant keywords among the initial candidates.

5.2.7 Selection of output

In the last step, we need to decide which of the final keywords should be displayed to the user. This is again influenced by the target application. In case of tag hinting, it is better to focus on recall and offer more keywords; although some of them will probably be irrelevant, the user can easily discard them. On the other hand, for direct annotation or evaluation

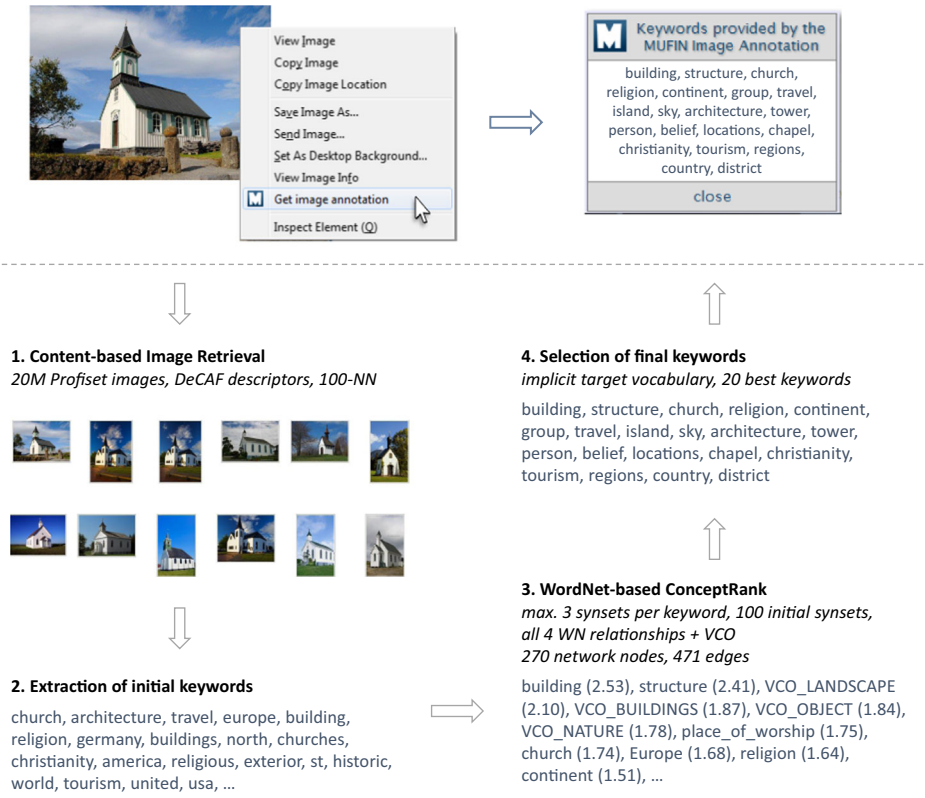


Fig. 6 Real image annotation performed by the MUFIN Image Annotation extension for Firefox

tasks we need to consider precision as well as recall. In this case, a smaller answer set is preferable.

Similar to most existing search-based systems, we select a fixed number of keywords with the highest probability of relevance as the annotation output. This simple approach reflects the semantics of the keyword probabilities, which express relative importance of the given keyword with respect to the query image rather than some absolute probability of the keyword relevance.

5.3 MUFIN image annotation demo

The functionality offered by the MUFIN Image Annotation tool is publicly available in the form of a web demo and a Firefox extension, which can be used to annotate arbitrary web images.[3] The complete annotation process with real-world data is recapitulated in Fig. 6, which shows intermediate results of individual phases as well as the specific parameter settings used in the example. We have also created a prototype application that utilizes content-based retrieval and the MUFIN Image Annotation to create visual and textual summaries of image collections [4].

[3] http://disa.fi.muni.cz/prototype-applications/image-annotation/

6 Experimental evaluation

To assess the effectiveness and efficiency of both the ConceptRank algorithm and the MUFIN Image Annotation system as a whole, we designed and evaluated two series of experiments. The objective of the first series was to analyze the influence of individual parameters on the annotation system performance. In the second series, the best-performing setting of our system was compared to other state-of-the-art image annotation techniques. In the following sections, we describe our methodology and analyze experimental results for both series.

6.1 Qualitative analysis

For the evaluation of the MUFIN Image Annotation tool performance with different parameter settings, we utilized the development data provided within the ImageCLEF 2014 Scalable Concept Image Annotation challenge [50]. This challenge was designed to compare annotation techniques that do not require precisely labeled training data, and are able to work with variable target vocabularies.

6.1.1 Data and metrics

The Scalable Concept Image Annotation (SCIA) task is defined by a binary input image and a target vocabulary that contains a list of eligible concepts (defined by WordNet synsets). A sample query is depicted in Fig. 7. During the development of their solutions, the participants of the SCIA challenge could use a development set of 1940 queries, for which a ground truth is available. This set was utilized in our experiments. There are 107 different target concepts in the development set, with the size of individual vocabularies ranging from 40 to 107 concepts.

Since the SCIA challenge focused in particular on the concept-wise scalability of annotation techniques, participants were not given any hand-labeled training data. However, a small training dataset was provided by the organizers, consisting of 500K web images located by a text search using the target concepts. Each image is accompanied by keywords extracted from the respective web page. This dataset (denoted as the *SCIA dataset*) was also used in some of our experiments.

To assess the quality of annotations, we adopted the full scope of SCIA quality measures [50]: mean precision (MP), mean recall (MR), mean F-measure (MF), and mean average precision (MAP). All these measures can be computed from two different perspectives: concept-based (denoted as MP-c, etc.) and sample-based (MP-s). A concept-based

aerial airplane baby beach bicycle bird boat bridge building car cartoon castle cat chair child church cityscape closeup cloud cloudless coast **countryside** **daytime** desert diagram dog drink drum elder embroidery fire firework fish flower fog food footwear furniture garden **grass** guitar harbor hat helicopter highway **horse** indoor instrument lake lightning logo monument moon motorcycle mountain nighttime overcast painting park person **plant** portrait protest rain rainbow reflection river road sand sculpture sea shadow sign silhouette smoke snow soil space spectacles sport sun sunrise/sunset table teenager toy traffic train tricycle truck underwater unpaved wagon water

Fig. 7 ImageCLEF query image and target vocabulary; ground truth concepts are highlighted

precision (or any other measure) is computed for each target concept, whereas a sample-based precision is computed for each test image. In both cases, the arithmetic mean is used as a global measure of performance. The annotation tool efficiency is measured by average query processing time.

6.1.2 Discussion of results

As discussed in Sections 4 and 5, most MUFIN Image Annotation tool components are parametrized by variables. In Table 3, we provide an overview of the most important parameters, which were studied in the experiments. The table also shows which values were examined in the experiments, and the best settings that were identified. In the following text, we focus on the most important findings and trends.

6.1.3 Content-based retrieval of similar images

In the similarity search phase, we analyzed the influence of two factors: the size and quality of the annotated image collection (AIC), and the number k of the most similar images retrieved by CBIR. To assess the influence of AIC properties, we experimented with various subsets of the Profiset and the SCIA dataset (see Fig. 8). As expected, the annotation quality grows with the increasing size and quality of the AIC. The unsupervised SCIA collection provides much noisier information than the supervised Profiset and by itself is not suitable for image annotation. However, the SCIA dataset does slightly increase the annotation quality when the two datasets are combined, because it covers all target concepts of the SCIA queries. Overall, it can be assumed that the quality of search-based annotation will further grow as larger, high-quality AIC become available.

The optimal number of nearest neighbors can only be determined for a specific dataset, visual descriptors, and distance measure, as it is influenced by the density of the similarity space and the quality of the visual similarity measures. In our case, when both the descriptor quality and dataset quality are high, the optimal number of visual neighbors is between 50 and 100. As we can observe in both Figs. 8 and 9, when we increase the number of

Table 3 Overview of annotation parameters and tested values

Annotation phase	Parameter	Tested values	Best-performing
CBIR	AIC type	SCIA dataset, Profiset	Profiset
	AIC size	500K – 20M	20M
	k (# of similar images)	20 – 200	100
Semantic network construction	Initial keyword scores	None / frequency-based / distance-based	Distance-based
	# of synsets/keyword	1 – unlimited	3 or 5
	# of initial synsets	50 – unlimited	300
	Types of relationships	Various combinations of hypernym, hyponym, holonym, and meronym	All relationships
ConceptRank computation	Relationship weights	Various	Same weight for all relationships
	Restart probability	0 – 0.4	0.2
Selection of final keywords	# of output keywords	1 – 30	5

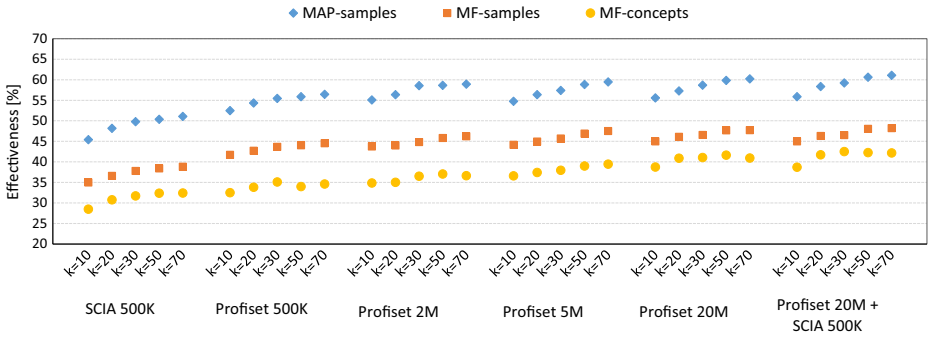


Fig. 8 Influence of the reference dataset properties and the number of nearest neighbors

visual neighbors the annotation quality first grows, then it stabilizes, and finally it begins to decrease again as less relevant images begin to appear in the CBIR result.

For the sake of readability, Figs. 8 and 9 (as well as most of the following tables and graphs) depict only selected quality measures. However, the observed trends were consistent for all metrics. The quality of results was significantly better from the sample-based perspective than from the concept-based one, which could be expected since the MUFIN Image Annotation tool was designed for hinting relevant keywords rather than checking the relevance of all vocabulary concepts. Moreover, some of the SCIA concepts are quite difficult to recognize (e.g. *unpaved* or *elder*).

6.1.4 Candidate keyword processing by ConceptRank

For the ConceptRank semantic analysis, the most important factors are 1) the number of initial synsets in the semantic network, 2) the types of relationships that form the network edges, and 3) the inclusion of visual-based relevance clues into the random walk computation.

As discussed in Section 4.4, we limit the number of initial synsets in order to cut down the annotation processing costs. From Fig. 9 we can conclude that it is not necessary to employ more than 300 initial synsets to obtain the maximum annotation quality – clearly the initial synsets with very low probability of relevance do not contribute any useful information. When the annotation speed is a priority, it is sufficient to use 100 synsets, which significantly reduces the costs but still provides good results.

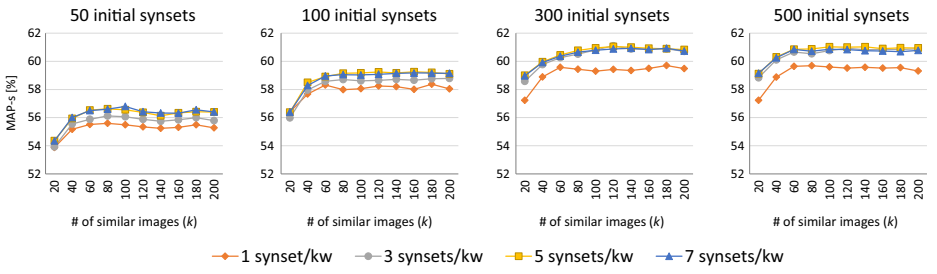


Fig. 9 Influence of the number of initial synsets

Table 4 Influence of relationships employed within ConceptRank (queries with large vocabularies)

Hypernym	Hyponym	Holonym	Meronym	MP-s	MR-s	MF-s	MAP-s	# of nodes / edges
–	–	–	–	42.68	52.79	44.71	53.72	300 / 0
enrich	–	–	–	44.53	54.09	46.28	56.46	300 / 37
EXPAND	–	–	–	46.74	56.84	48.64	60.50	668 / 603
EXPAND	enrich	–	–	46.83	56.89	48.72	60.51	668 / 1139
EXPAND	enrich	enrich	enrich	**48.00**	**58.15**	**49.91**	**61.32**	668 / 1206

The best result for each quality measure is highlighted by bold face

As for the types of semantic relationships, our experiments show that the most significant improvement is brought by the hypernymy hierarchy (see Table 4). The other relationships only improve the annotation quality marginally, but their processing does not add any significant time overhead. The influence of semantic relationships is most pronounced for queries with large vocabularies, where the overlap between initial candidate keywords and the target vocabulary is high. To achieve the best results, the hypernymy relationship should be used in the expansion mode, so that new nodes are added to the semantic network during the exploration of hypernyms. All other relationships work better in the enrichment mode, e.g. only adding edges between existing network nodes. The best results were observed when the relative importance of all relationships was the same. For the bottom-up relationships (hypernymy and holonymy), each edge was assigned a constant weight; for the top-down relationships, this constant was distributed among all outgoing edges of a given node.

To evaluate the importance of the input visual scores of the candidate keywords, we also conducted several experiments that did not take the visual information into consideration during the ConceptRank computation. Let us remember that the visual scores of initial keywords are transformed to non-zero initial probabilities of individual network nodes, which determine the biased restart vector for the random walk. The visual scores thus provide two types of information: which nodes are the initial ones that may be targeted by the restart, and what is the probability that the random walk will restart in a given node. If we decide to ignore the initial probabilities, we may still use the information about which nodes were the initial ones, or we can opt to ignore this as well. Accordingly, we evaluated two variations of the ConceptRank algorithm without the visual scores. In the *UniformBiasedRestart* variant, we keep the information about the initial nodes; only these nodes can be targets of the restart, which is thus again biased, but the restart probability is the same for each initial node. On the contrary, the *UniformRestart* variant does not distinguish between initial and other nodes, so that the whole network is processed by a standard random walk where all nodes are equally likely to be a target of the random restart. Table 5 shows the results

Table 5 Importance of visual clues within the ConceptRank

Candidate keyword processing method	MP-s	MR-s	MF-s	MAP-s
Most frequent keywords without ConceptRank	42.68	52.79	44.71	53.72
ConceptRank-UniformRestart, all relationships	40.67	49.80	42.52	54.68
ConceptRank-UniformBiasedRestart, all relationships	45.87	56.19	47.73	58.02
ConceptRank with visual clues, all relationships	**48.00**	**58.15**	**49.91**	**61.32**

The best result for each quality measure is highlighted by bold face

for both these variations compared to the original ConceptRank. As we can see, even if the initial visual scores are removed, the semantic links improve the quality of annotation result significantly. However, adding the information about initial nodes and restart probabilities gradually improves the results. The best results are achieved when the semantic relationships are combined with the full visual-based information expressed by initial node probabilities, which confirms our expectations about the synergy between the visual and semantic relevance.

From Table 3, we can further observe that better results were achieved when the initial candidate keyword scores were based on the respective images' visual distances rather than plain frequencies. For each keyword, it is helpful to consider several possible synsets. The optimal restart probability for the ConceptRank random walk seems to be 0.2, but the differences in result quality were minimal for all restart values larger than 0.05. Only the very low restart coefficients suppressed the influence of initial synset probabilities from the CBIR phase, which resulted in significantly worse annotations.

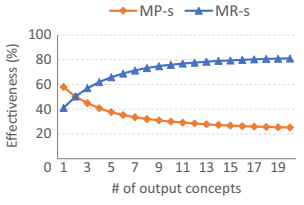
6.1.5 Selection of output

The final step of the annotation is the selection of the output keywords. As discussed in Section 5, we currently select a fixed number of the most probable keywords from the target vocabulary. This number needs to be selected with respect to the target application: if we aim at stand-alone annotation, we need to balance the precision and recall. On the other hand, for the keyword hinting scenario we are more interested in recall, i.e. the number of relevant keywords that the application offer.

In all the previous evaluations, the annotation result was formed by at most 5 most probable keywords (less if there were not enough candidates with non-zero probability). The constant 5 was chosen as a reasonable trade-off between the result precision and recall; in Fig. 10, we depict the sample-based precision and recall for different sizes of the result set. Naturally, these numbers are only valid for the ImageCLEF task, since the optimal result size is influenced by the average size of the ground truth. Actually, using the fixed answer size is not the most suitable strategy for the ImageCLEF data, since the size of ground truth is quite variable here (1–11 keywords). However, even with this simple approach to the final result selection we achieved very good results in the ImageCLEF competition, as will be demonstrated in the following section.

6.1.6 Efficiency

Since our research aims at real-time keyword hinting, it is also highly important whether the annotation processing is efficient. In Table 6, we present the computation times of



Concepts with non-zero probability, ordered by descending probability. Concepts that are correct according to GT are highlighted. **water**; **bridge**; **sky**; building; cityscape; **river**; road; cloud; sea; lake; sunrise/sunset; highway; painting; coast; beach; vehicle; boat; car; grass; harbor; traffic; sun; train; **plant**; forest; female

Missing GT concepts: cloudless; flower

Fig. 10 Precision and recall for various result set sizes (*left*); sample annotation result with highlighted ground-truth concepts (*right*)

Table 6 Processing costs of individual annotation phases

Phase	Time [ms]		Time complexity
	100 init. s.	300 init. s.	
DeCAF descriptor extraction	45	45	– constant
CBIR in 20M images (using PPP-Codes index)	390	390	– sublinear to dataset size – influence of k is negligible
Processing of keywords for the most similar images	20	20	– linear to the number of keywords per image
Semantic network construction (different combinations of relationships)	21–25	24–27	– quadratic to the number of initial synsets – linear to the number and type of relationships
ConceptRank computation (approximate/precise)	47 / 770	205 / 6128	– cubic to the number of network nodes and edges (precise) – linear to the number to the number of network nodes and edges (approximate)

individual annotation phases. With the most efficient settings, each image takes about 0.6 second to process. As we can see, most of the time is spent on the similarity searching, while the semantic analysis is very fast when we employ the described approximations. Most important of these is the approximation of the ConceptRank computation (see Section 4.4 for details); limiting the number of matrix computations has a negligible impact on result quality while the costs reduction is essential.

We can observe that for the presented settings, we have achieved the desired real-time response of the annotation system. Moreover, the overall time complexity is most influenced by the CBIR phase, which scales sublinearly [37]. The other parts of the processing are either very cheap (processing of keywords, semantic network construction) or can be efficiently approximated without significant impact on annotation quality (ConceptRank computation). Therefore, the MUFIN Image Annotation tool can be expected to scale well and allow interactive operation even for significantly larger reference datasets. It is also worth noticing that the size of the output vocabulary does not influence the complexity of any processing phase. The MUFIN Image Annotation tool therefore scales perfectly with respect to the target vocabulary size.

6.2 Comparison with other approaches

The comparison to state-of-the-art annotation techniques was performed on two platforms. First, we participated in the ImageCLEF 2014 Scalable Concept Image Annotation challenge, which was described in the previous section. This allowed us to compare our tool to others in a well-defined contest that is well-known in the image annotation field. Second, we created our own testbed for the evaluation of annotation queries with unlimited vocabularies. On this testbed, we evaluated the usefulness of our tool in the context of general web image annotation, and compared its results to the Google Vision API.

6.2.1 ImageCLEF 2014 scalable concept image annotation challenge

6.2.2 Data and metrics

As described earlier, the Scalable Concept Image Annotation (SCIA) task is defined by a query image and a set of eligible concepts, and the objective is to identify all relevant concepts. The actual SCIA competition consisted of annotating 7291 images with different concept lists. Altogether, there were 207 concepts, with the size of individual concept lists ranging from 40 to 207 concepts. Let us remember here that the development set contained only 107 concepts; the other 100 were not available beforehand in order to test the concept-wise scalability of individual competing techniques. The set of quality measures was already introduced in Section 6.1.1.

6.2.3 Discussion of results

Our participation in the ImageCLEF 2014 SCIA competition is described in detail in [8]. We entered the competition under the name DISA, referring to the name of our lab. Basically, we utilized the MUFIN Image Annotation tool as described in Section 5. The only significant difference was in the CBIR phase, where we employed different visual descriptors to search for similar images. In particular, we utilized a combination of five MPEG7 descriptors as described in [31]. With the MPEG7 descriptors, our submission placed as 5th among 11 participants of the contest. However, shortly after the competition deadline we finished a new implementation of the CBIR with the state-of-the-art DeCAF descriptors. The SCIA competition organizers kindly evaluated also these new results, although out of contest. As shown in Table 7, the change of visual descriptors significantly improved the annotation quality of our tool and moved the DISA up to the 2nd position. A detailed description about the differences between the MPEG7 and DeCAF implementation can be found in [9].

Table 7 The SCIA competition results table from [50] with a new line for DISA DeCAF results

System	MAP-samples				MF-samples				MF-concepts				
	all	ani.	food	207	all	ani.	food	207	all	ani.	food	207	unseen
KDEVIR 9	36.8	33.1	67.1	28.9	37.7	29.9	64.9	32.0	54.7	67.1	65.1	31.6	66.1
DISA DeCAF	**48.7**	**51.0**	**67.1**	**32.3**	**39.9**	**44.4**	**48.5**	**26.7**	**41.1**	**45.3**	**42.1**	**22.4**	**44.9**
MIL 3	36.9	30.9	68.6	23.3	27.5	20.6	53.1	18.0	34.7	34.7	50.4	16.9	36.7
MindLab 1	37.0	43.1	63.0	22.1	25.8	17.0	45.2	18.3	30.7	35.1	35.3	16.7	34.7
MLIA 9	27.8	18.8	53.6	16.7	24.8	12.1	46.0	16.4	33.2	32.7	37.3	16.9	34.8
DISA 4	**34.3**	**46.6**	**39.6**	**19.0**	**29.7**	**40.6**	**31.2**	**16.9**	**19.1**	**23.0**	**22.3**	**7.3**	**19.0**
RUC 7	27.5	25.2	44.2	15.1	29.3	28.0	28.2	20.7	25.3	20.1	23.1	10.0	18.7
IPL 9	23.4	30.0	48.5	18.9	18.4	20.2	29.8	17.5	15.8	15.8	33.3	12.5	22.0
IMC 1	25.1	35.7	35.6	12.9	16.3	14.3	21.0	10.9	12.5	10.2	15.1	6.1	11.2
INAOE 5	9.6	6.9	15.0	8.5	5.3	0.4	0.5	6.4	10.3	1.0	0.8	17.9	19.0
NII 1	14.7	23.2	22.0	4.6	13.0	18.9	18.7	4.9	2.3	3.0	2.1	0.9	1.8
FINKI 1	6.9	N/A	N/A	N/A	7.2	8.1	12.3	4.1	4.7	6.3	9.0	2.9	4.7

Only the best result for each group is given. The systems are ranked by overall performance as defined in [50]

In comparison with the other competing groups, our solution ranked high in both sample-based mean F-measure and sample-based MAP. In particular, the sample-based MAP achieved by our DeCAF submission was the highest of all. The results for concept-based F-measure were less competitive, which did not come as a surprise. In general, the search-based approach works well for frequent terms, whereas concepts for which there are few examples are difficult to recognize. Table 7 also shows the annotation quality for selected subsets of the target queries and concepts. We can see that our solution did not return worse results for concepts that were not seen in the development set, which supports our claim at concept-based scalability of our approach.

The key factors of the MUFIN Image Annotation success in the SCIA competition were the DeCAF-based visual similarity, and the ConceptRank semantic analysis. The usefulness of the DeCAF descriptors is evident from the comparison of with MPEG7-based annotation in Table 7. To clarify the importance of the ConceptRank analysis, it is important to mention that several of the SCIA challenge participants also used some adaptation of neural network classifiers to annotate the images (more details can be found in [50]). The DISA DeCAF submission achieved better results than these groups, which confirms the importance of the semantic analysis step developed by our group. When we analyzed the usefulness of individual components of our solution, we found that the ConceptRank component improved the overall effectiveness by 5–10%, depending on the quality metric [8].

6.2.4 Profiset evaluation

The ImageCLEF competition focused on annotation tasks with changing but limited vocabularies, which may appear in various classification tasks. However, the primary target application for our annotation tool is open-vocabulary tag hinting, which requires test queries with unlimited vocabularies. Since we could not find any existing benchmark suitable for the tag-hinting scenario, we designed a new evaluation dataset using selected images from the Profiset collection.

6.2.5 Data and metrics

The Profiset evaluation dataset contains 160 images from the Promedia image-stock collections. 80 photos were selected from Promedia search logs of popular queries, another 80 were chosen randomly from images sold in the last two years. These images were removed from the Profiset collection, so there is no overlap between the test queries and the annotated image collection used as knowledge base in the CBIR phase of the MUFIN Image Annotation processing.

To be able to evaluate annotation tasks with unlimited vocabularies, we should further provide a ground truth of all English keywords relevant for a given image. However, this is hardly feasible, since there may be literally a thousand words describing each picture. Therefore, we did not attempt to collect the complete ground truth for our queries. Instead, we only assessed the relevance of keywords that were proposed by any of the annotation methods under evaluation, thus creating a partial ground truth sufficient for our comparison.

The relevance assessment was done in the following way. For each query, we collected all keywords suggested by all the methods under comparison. The keywords were merged and displayed to human assessors, who classified them as *highly-relevant*, *relevant* (the keyword represents some less important or less precise information), or *irrelevant*. The assessments were distributed among five dedicated persons, and each query was evaluated by at least two persons. The verbal assessments were then transformed to relevance scores of 1, 0.5

and 0, respectively. The final relevance of each keyword was computed as an average of the relevance scores. We decided to define two partial ground truths, using different threshold values for the relevant keywords: in *GT-HR*, there are only keywords with average relevance score equal to or greater than 0.75, whereas for *GT-R* the threshold score is 0.5.

Using *GT-HR* and *GT-R*, we can fairly evaluate the precision of individual methods, but the partial ground truth is not sufficient for measuring the recall. However, we have one more ground truth that can be used for this purpose – the original keyword descriptions of images from the Profiset collection, as provided by the image authors. This data, which we shall denote as *ProfisetGT*, is independent of the methods under comparison, so it is fair to evaluate both the precision and recall of individual approaches with respect to *ProfisetGT*.

6.2.6 State-of-the-art competition: google image annotation

As a representative of state-of-the-art open-vocabulary image annotation, we chose the Label Detection service provided by the Google Vision API. The Google Vision API[4] is a commercial image recognition system, which attempts to understand the content by employing powerful machine learning models. It allows to classify images into thousands of categories, detect individual objects and faces within images, and find and read printed words contained within images. The service was started at the end of 2015 and it is offered as cloud REST API, thus allowing Google to continuously improve the quality of the recognition as the research in this area advances.

Even though the specific methods are not disclosed, the system probably uses some variation of the Inception deep neural network model [45]. The Label Detection service claims to be able to recognize broad sets of categories within an image, ranging from modes of transportation to animals.

6.2.7 Discussion of results

Using the Profiset test queries, we have evaluated the quality of the MUFIN Image Annotation tool with the best-performing settings (see Section 6.1), and the Google Label Detection service. In both cases, we asked for at most 50 most probable descriptive keywords. Since there is currently no minimum relevance threshold for the MUFIN Image Annotation, it always returned the 50 keywords. Google Label Detection, however, returns only keywords with a minimum relevance probability of 50%. Consequently, the number of keywords returned by Google was significantly lower; there were no more than 20 keywords found for a single image, and the average annotation size was 5.7 keywords. For 11 queries, no keywords were returned at all by the Google Label Detection.

The precision and recall of both approaches is shown in Fig. 11. We can see how the result quality changes for different sizes of the annotation output. On the first positions of the annotation result, MUFIN Image Annotation is about 8% better in terms of precision. When the result size grows, Google Label Detection does not provide any new keywords, therefore its precision becomes constant. MUFIN Image Annotation continues to offer less probable keywords, which decreases the average precision of the result. At the same time, however, the total number of relevant keywords that were found grows, which is demonstrated by the recall curve.

[4] https://cloud.google.com/vision/

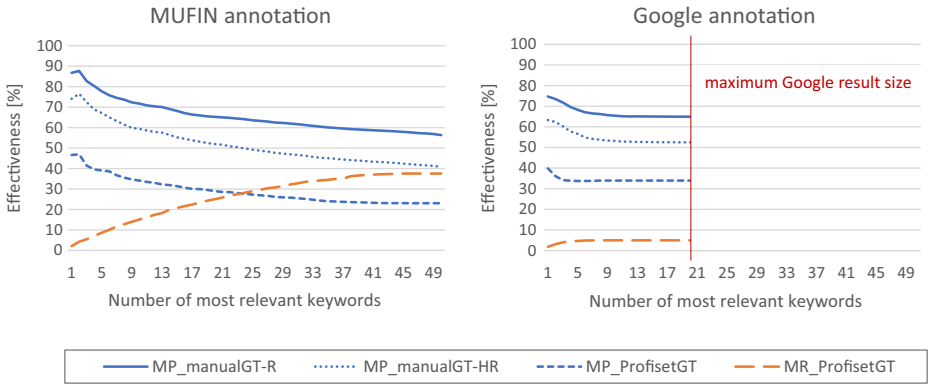


Fig. 11 Precision and recall of MUFIN and Google annotations

In Fig. 12, we can see three test queries and the respective annotation results, which were selected to highlight the strengths and weaknesses of both techniques. For some images, Google provides perfect keywords (the last query in Fig. 12), but for others it completely fails (the first image in Fig. 12). For comparison, MUFIN Image Annotation failed to return anything relevant among the top 5 keywords only for 3 images. The second query in Fig. 12 is an example of a situation where both techniques work well; we can observe here the differences in the type and scope of keywords returned by the two approaches. We have also computed the average overlap between the two annotations; on average, 1.9 keywords appear in both results, out of which 1.75 keywords is relevant.

Overall, the MUFIN Image Annotation returned better results than Google Label Detection in our experiments. If we limit the number of MUFIN Image Annotation results to 10, both the precision and recall are higher than in case of Google Label Detection. MUFIN

Google keywords
product

MUFIN keywords
person, **adult**, animals, activity, **scientist**, woman, knowledge, **people**, health, **research**, work, wellbeing, **science**, **indoors**, **one**, **laboratory**, **head**, mid, **years**, **man**, **clothing**, female, medical, doctor, coat, care, prosperity, hospital, **men**, worker, male, think, **equipment**, personnel, technician, **researcher**, working, young, professionals, color, **occupations**, technology, bioscience, organization, two, **photography**, healthcare, holding, african

Google keywords
penguin, flightless_bird, vertebrate, **bird**

MUFIN keywords
penguin, **animal**, **group**, **bird**, seabird, aptenodytes, wildlife, chicks, **snow**, **continent**, **baby**, **emperor**, children, hill, offspring, **young**, island, **outdoors**, **sea**, ice, **water**, **nobody**, birds, daytime, cold, weather, nature, color, **flightless**, wild, laughingstock, colony, adult, day, glacier, fauna, outdoor, body, polar, travel, **photography**, marine, antarctic, horizontal, region, natural, peninsula, cute, outside, regions

Google keywords
bumper, automotive_design, automotive_exterior, vehicle, car, wheel, land_vehicle, sports_car, mercedes_benz, supercar, automobile_make, mercedes_benz_slr_mclaren, model_car

MUFIN keywords
car, show, vehicle, travel, transport, sports, motor, automobile, speed, person, **luxury**, coupe, **new**, museum, road, indoors, concept, color, view, manufacturers, front, three, automotive, horizontal, **expensive**, nobody, convertible, business, photography, roadster, industry, european, study, transportation, **fast**, photo, silver, **modern**, salon, make, street, white, showpiece, cars, black, republic, city, studio, district, state

Fig. 12 Several examples of annnotation results returned by Google and MUFIN. The Profiset GT is highlighted by *bold face*, manual GT-R by *underlining*

Image Annotation is also able to continuously provide additional candidates, until the user is satisfied. On the other hand, the user then has to discard an increasing number of irrelevant candidates. From this perspective, the smaller size of Google Label Detection results can be considered an advantage. However, the Google probability threshold is likely too strict, since it sometimes does not allow rather obvious concepts into the result (e.g. *person* in the first query in Fig. 12).

7 Conclusions and future work

In this paper, we have studied the topic of search-based image annotation, which is a complex problem that consists of several challenging subtasks, including the acquisition of a suitable knowledge base, efficient and effective CBIR, and the semantics mining in candidate keywords. We have demonstrated that by combining state-of-the-art techniques available for individual subtasks, we can create a search-based annotation tool that succeeds in comparison with other academic prototypes based on different principles, as well as with a state-of-the-art commercial solution.

The main theoretical contribution of the paper is the ConceptRank algorithm, which allows us to semantically connect candidate keywords and identify the most probable content of the query object. The algorithm is applicable to any data domain, can be efficiently implemented, and scales well with respect to both the knowledge base size and the target vocabulary size. Its effectiveness was demonstrated within the MUFIN Image Annotation tool, where we were able to increase the mean average precision of annotations by approximately 10%.

In the future, the presented work can be continued in several directions. As we outlined in Section 2, the search-based approach to multimedia annotation is in many aspects complementary to traditional classifiers. To exploit the strengths of both approaches, we would like to explore possible combinations of the ConceptRank model and the classifier outputs. Another input for the annotation can be provided by users in the form of relevance feedback, which could also be integrated into the ConceptRank model. It would also be interesting to implement the ConceptRank analysis with more semantic resources, e.g. ontologies or alternative language models such as word2vec.

Acknowledgements This paper is based on research supported by the Czech Science Foundation project No. P103/12/G084.

References

1. Bartolini I, Ciaccia P (2010) Multi-dimensional keyword-based image annotation and search. In: Proceedings of the 2nd International workshop on keyword search on structured data (KEYS 2010), pp 5:1–5:6
2. Batko M, Novak D, Zezula P (2007) MESSIf: Metric similarity search implementation framework. In: 1st international DELOS conference, revised selected papers. Springer, LNCS, vol 4877, pp 1–10
3. Batko M, Botorek J, Budikova P, Zezula P (2013) Content-based annotation and classification framework: a general multi-purpose approach. In: 17th international database engineering & applications symposium (IDEAS 2013), pp 58–67
4. Batko M, Budikova P, Elias P, Zezula P (2014) CLAN Photo presenter: Multi-modal summarization tool for image collections. In: International conference on multimedia retrieval (ICMR 2014), pp 541–542
5. Botorek J, Budikova P, Zezula P (2014) Visual concept ontology for image annotations. CoRR arXiv: 1412.6082

6. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw 30(1-7):107–117
7. Budikova P, Batko M, Zezula P (2011) Evaluation platform for content-based image retrieval systems. In: International conference on theory and of digital libraries (TPDL 2011), pp 130–142
8. Budikova P, Botorek J, Batko M, Zezula P (2014) DISA at ImageCLEF 2014: The search-based solution for scalable image annotation. In: CLEF 2014: evaluation labs and workshop, Online Working Notes
9. Budikova P, Batko M, Botorek J, Zezula P (2015) Search-based image annotation: Extracting semantics from similar images. In: Experimental IR meets multilinguality, multimodality, and interaction: 6th international conference of the CLEF association (CLEF 2015), pp. 327–339
10. Cai X, Wang H, Huang H, Ding CHQ (2012) Simultaneous image classification and annotation via biased random walk on tri-relational graph. In: 12th European conference on computer vision (ECCV 2012), pp 823–836
11. Caputo B, Müller H, Martinez-Gomez J, Villegas M, Acar B, Patricia N, Marvasti N, ÜSküdarlı S, Paredes R, Cazorla M, Garcia-Varea I, Morell V (2014) ImageCLEF2014: Overview and analysis of the results. In: CLEF proceedings, lecture notes in computer science. Springer, Berlin Heidelberg
12. Dai L, Wang X, Zhang L, Yu N (2012) Efficient tag mining via mixture modeling for real-time search-based image annotation. In: Proceedings of the 2012 IEEE international conference on multimedia and expo (ICME 2012), pp 134–139
13. Deng J, Dong W, Socher R, Li LJ, Li K, Li FF (2009) ImageNet: A large-scale hierarchical image database. In: IEEE computer society conference on computer vision and pattern recognition (CVPR 2009), pp 248–255
14. Deselaers T, Keysers D, Ney H (2008) Features for image retrieval: an experimental comparison. Inf Retr 11:77–107
15. Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2014) DeCAF: a deep convolutional activation feature for generic visual recognition. In: Proceedings of the 31th international conference on machine learning (ICML 2014), pp 647–655
16. Fellbaum C (1998) WordNet: An electronic lexical database. The MIT Press
17. Fu J, Wang J, Rui Y, Wang X, Mei T, Lu H (2015) Image tag refinement with view-dependent concept representations. IEEE Trans Circ Syst Video Technol 25(8):1409–1422
18. Gupta MR, Bengio S, Weston J (2014) Training highly multiclass classifiers. J Mach Learn Res 15(1):1461–1492
19. Gyöngyi Z, Garcia-Molina H, Pedersen J (2004) Combating web spam with TrustRank. In: Proceedings of the 30th international conference on very large data bases - volume 30, VLDB Endowment, VLDB '04, pp 576–587
20. He X, Li X, Yang G, Xu J, Jin Q (2014) Adaptive tag selection for image annotation. In: Advances in multimedia information processing (PCM 2014), pp 11–21
21. Hu J, Lam KM (2013) An efficient two-stage framework for image annotation. Pattern Recogn 46(3):936–947
22. Ke X, Li S, Chen G (2013) Real web community based automatic image annotation. Comput Electr Eng 39(3):945–956
23. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems (NIPS 2012), pp 1106–1114
24. Leskovec J, Rajaraman A, Ullman JD (2014) Mining of massive datasets, 2nd edn. Cambridge University Press, Cambridge
25. Li X, Chen L, Zhang L, Lin F, Ma W (2006) Image annotation by large-scale content-based image retrieval. In: Proceedings of the 14th ACM international conference on multimedia, pp 607–610
26. Lin Z, Ding G, Hu M, Wang J, Sun J (2012) Automatic image annotation using tag-related random search over visual neighbors. In: 21st ACM international conference on information and knowledge management (CIKM'12), pp 1784–1788
27. Lin Z, Ding G, Hu M (2015) Image auto-annotation via tag-dependent random search over range-constrained visual neighbours. Multimedia Tools Appl 74(11):4091–4116
28. Lindstaedt SN, Mörzinger R, Sorschag R, Pammer V, Thallinger G (2009) Automatic image annotation using visual content and folksonomies. Multimedia Tools Appl 42(1):97–113
29. Liong VE, Lu J, Wang G, Moulin P, Zhou J (2015) Deep hashing for compact binary codes learning. In: IEEE conference on computer vision and pattern recognition, CVPR 2015, pp 2475–2483
30. Liu D, Hua X, Yang L, Wang M, Zhang H (2009) Tag ranking. In: Proceedings of the 18th international conference on world Wide Web (WWW 2009), pp 351–360
31. Lokoc J, Novák D, Batko M, Skopal T (2012) Visual image search: Feature signatures or/and global descriptors. In: 5th international conference on similarity search and applications (SISAP 2012), pp 177–191

32. Lux M, Pitman A, Marques O (2010) Can global visual features improve tag recommendation for image annotation? Future Internet 2(3):341–362
33. Maier O, Kwasnicka H, Stanek M (2012) Image auto-annotation with automatic selection of the annotation length. J Intell Inf Syst 39(3):651–685
34. Makadia A, Pavlovic V, Kumar S (2008) A new baseline for image annotation. In: 10th European conference on computer vision (ECCV 2008), pp 316–329
35. Muja M, Lowe DG (2014) Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans Pattern Anal Mach Intell 36(11):2227–2240
36. Naidan B, Boytsov L, Nyberg E (2015) Permutation search methods are efficient, yet faster search is possible. In: Proceedings of the 41st international conference on very large data bases, pp 1618–1629
37. Novak D, Zezula P (2016) PPP-codes for large-scale similarity searching. Transactions on Large-Scale Data- and Knowledge-Centered Systems XXIV 9510:61–87
38. Novak D, Batko M, Zezula P (2015) Large-scale image retrieval using neural net descriptors. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp 1039–1040
39. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
40. Schickel-Zuber V, Faltings B (2007) OSS: A semantic similarity function based on hierarchical ontologies. In: Proceedings of the 20th international joint conference on artificial intelligence, pp 551–556
41. Sivic J, Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. In: 9th IEEE international conference on computer vision (ICCV 2003), pp 1470–1477
42. Smeulders A, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380
43. Suchanek FM, Kasneci G, Weikum G (2008) YAGO: A large ontology from wikipedia and wordnet. J Web Semantics 6(3):203–217
44. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition (CVPR 2015), pp 1–9
45. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2015) Rethinking the inception architecture for computer vision. CoRR arXiv: 1512.00567
46. Tong H, Faloutsos C, Pan J (2008) Random walk with restart: fast solutions and applications. Knowl Inf Syst 14(3):327–346
47. Torralba A, Fergus R, Freeman WT (2008) 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Trans Pattern Anal Mach Intell 30(11):1958–1970
48. Tousch AM, Herbin S, Audibert JY (2012) Semantic hierarchies for image annotation: a survey. Pattern Recogn 45(1):333–345
49. Vijayanarasimhan S, Shlens J, Monga R, Yagnik J (2014) Deep networks with large output spaces. CoRR arXiv: 1412.7479
50. Villegas M, Paredes R (2014) Overview of the ImageCLEF 2014 scalable concept image annotation task. In: Working notes for CLEF 2014 conference, pp 308–328
51. Wang C, Jing F, Zhang L, Zhang H (2006) Scalable search-based image annotation of personal images. In: Proceedings of the 8th ACM SIGMM international workshop on multimedia information retrieval (MIR 2006), pp 269–278
52. Wang C, Jing F, Zhang L, Zhang H (2008) Scalable search-based image annotation. Multimedia Syst 14(4):205–220
53. Wang C, Blei DM, Li F (2009) Simultaneous image classification and annotation. In: IEEE computer society conference on computer vision and pattern recognition (CVPR 2009), pp 1903–1910
54. Wang XJ, Zhang L, Ma WY (2012) Duplicate-search-based image annotation using web-scale data. Proc IEEE 100(9):2705–2721
55. Wang X, Du J, Wu S, Li X, Xin H, Zhang Y, Li F (2015) High-level semantic image annotation based on hot internet topics. Multimedia Tools Appl 74(6):2055–2084
56. Yu J, Cao D, Li S, Lin D (2012) A novel image annotation feedback model based on internet-search. In: Web information systems and mining (WISM 2012), pp 580–588

57. Zezula P, Amato G, Dohnal V, Batko M (2006) Similarity search: the metric space approach, advances in database systems, vol 32. Springer-Verlag
58. Zhang D, Islam MM, Lu G (2012) A review on automatic image annotation techniques. Pattern Recogn 45(1):346–362
59. Zhang L, Chen L, Li M, Zhang H (2003) Automated annotation of human faces in family albums. In: Proceedings of the 11th ACM international conference on multimedia, pp 355–358
60. Zhang X, Li Z, Chao WH (2013) Improving image tags by exploiting web search results. Multimedia Tools Appl 62(3):601–631

Petra Budikova is a researcher at the Faculty of Informatics, Masaryk University, Brno, Czech Republic, where she obtained a Ph.D. Degree in computer science in 2013. In her research, she focuses on multimodal analysis of multimedia data with the utilization of content-based retrieval techniques. She is mainly interested in the synergy between the image and text modalities, and their utilization for image retrieval and annotation.

Michal Batko is an assistant professor at the Faculty of Informatics, Masaryk University, Brno, Czech Republic, where he obtained a Ph.D. Degree in computer science. His research activities concentrate on the efficient searching in large distributed environments with emphasis on the scalability problem. The focus is especially on the problems of distributed similarity search using the metric space approach. As a software developer, he coordinates the development of an extensive similarity searching framework used for creating prototypes of indexing techniques and multimedia retrieval systems.

Pavel Zezula is a professor of informatics at the Faculty of Informatics, Masaryk University, Brno, Czech Republic. His professional interests concentrate on storage structures and algorithms for scalable content-based retrieval in non-traditional digital data types and formats. He has participated in numerous EU projects. His research team at the Masaryk University developed an extensible, scalable, and infrastructure independent similarity search engine. He is a co-author of more than 100 conference and 30 journal papers as well as the book Similarity Search: the Metric Space Approach, published by Springer US, 2006.