Combining Image Captions and Visual Analysis for Image Concept Classification

Tomas Kliegr
Department of Information and Knowledge Engineering
Faculty of Informatics and Statistics, University of Economics, Prague
tomas.kliegr@vse.cz

Krishna Chandramouli
Multimedia and Vision Research Group
Queen Mary University
Mile End Road, London
United Kingdom
krishna.c@ieee.org

Jan Nemrava
Department of Information and Knowledge Engineering
Faculty of Informatics and Statistics, University of Economics, Prague
nemrava@vse.cz

Vojtech Svatek
Department of Information and Knowledge Engineering
Faculty of Informatics and Statistics, University of Economics, Prague
svatek@vse.cz

Ebroul Izquierdo
Multimedia and Vision Research Group
Queen Mary University
Mile End Road, London
United Kingdom
ebroul.izquierdo@elec.qmul.ac.uk

ABSTRACT

We present a framework for efficiently exploiting free-text annotations as a complementary resource to image classification. A novel approach called Semantic Concept Mapping (SCM) is used to classify entities occurring in the text to a custom-defined set of concepts. SCM performs unsupervised classification by exploiting the relations between common entities codified in the Wordnet thesaurus. SCM exploits Targeted Hypernym Discovery (THD) to map unknown entities extracted from the text to concepts in Wordnet. We show how the result of SCM/THD can be fused with the outcome of Knowledge Assisted Image Analysis (KAA), a classification algorithm that extracts and labels multiple segments from an image. In the experimental evaluation, THD achieved an accuracy of 75%, and SCM an accuracy of 52%. In one of the first experiments with fusing the results of a free-text and image-content classifier, SCM/THD + KAA achieved a relative improvement of 49% and 31% over the text-only and image-content-only baselines.

1. INTRODUCTION

Images are often accompanied by free-text annotations, which describe what is on the image and thus can serve as a valuable complementary source of information for content-based image classifiers. The fact that annotations often refer to specific places and people (named entities) that appear on the image has, however, ironically hindered their utilization in image classification tasks, since existing approaches to integration of textual annotations with image content do not handle well uncommon words and particularly named entities. The use of existing systems for Named Entity Recognition (NER) is limited, because they only categorize named entities to several predefined classes, which is insufficient for the general image classification task.

In this paper we present a framework for efficiently exploiting free-text image annotations, with special focus on named entities. A novel approach called Semantic Concept Mapping (SCM) is used to classify entities occurring in the text to a custom-defined set of concepts (classes). SCM performs unsupervised classification by exploiting the relations between common entities codified in the Wordnet thesaurus.

SCM uses a new variation of hypernym discovery, called Targeted Hypernym Discovery (THD), to map an unknown entity extracted from the text to a concept in the Wordnet thesaurus. The most appropriate documents defining the entity are found in a large encyclopedic corpora and lexico-syntactic patterns are used to extract the hypernym.

In order to demonstrate the benefit of our approach, we show how the result of SCM can be fused with the outcome of a specific image classification algorithm. We chose Knowledge Assisted Image Analysis (KAA) [20], which considers a raw image as the input and produces a set of segments, each associated with a corresponding label from a predefined set of semantic concepts. The resulting experimental framework is depicted in Figure 1.

We experimentally evaluated the performance of THD, SCM and KAA. In the final experiment, the class predictions of SCM/THD were fused with the classification result of KAA.

Paper organization: The proposed approach to the analysis of text consisting of SCM and THD is described in Sections 2 and 3. Section 4 briefly describes the KAA im-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MDM/KDD'08, August 24, 2008, Las Vegas, NV, USA
Copyright 2008 ACM 978-1-60558-261-0 ...$5.00.

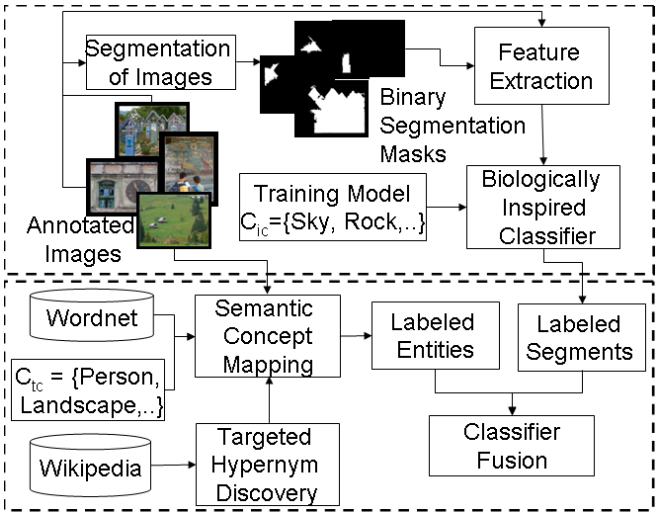

Figure 1: Framework overview

plementation used and the preliminary approach to classifier fusion. Experimental evaluation is presented in Section 5. Related research is presented in Section 6, followed by a discussion and an outline of future work in Section 7. Section 7 summarizes our contribution and provides conclusions.

2. SEMANTIC CONCEPT MAPPING

The goal of SCM is to facilitate the fusion of text analysis with the results of image classification. Image classification algorithms typically perform classification of whole images or image segments to multiple categories with the choice of categories varying from application to application. The classifier needs to be retrained each time the set of categories changes. This is not a problem since the amount of required training data is usually small enough to make the design of human-labeled training sets feasible. The resulting classifiers often have soft outputs, assigning a confidence measure to the prediction of class label.

If textual annotations were to be classified in a similar fashion as image classifiers, a much larger training set would be necessary. The reason being that when learning text classifiers, one often has to deal with very large numbers (more than 10,000) of features [16], since there is typically one feature for every word. For comparison, images are represented with much lower number of features, e.g. the KAA system introduced in section 4.1 only uses a 98-dimensional feature vector consisting of low-level image features such as color, texture, shape, etc. In addition, the feature vectors of individual annotations are extremely sparse; [28] reports a ratio of 158:1 between the average number of terms in an image caption and the total number of terms in the collection of captions. Data sparsity has a particularly severe effect on uncommon words including named entities, which are often of central importance in image annotations.

Since large training sets (annotated corpora) are generally unavailable and expensive to design for an ad hoc set of classes[1], with Semantic Concept Mapping (SCM) we propose to take a 'linguistic' rather then statistical approach to classification.

SCM proceeds in a similar way as a human evaluator would do if presented with an image annotation together with a pool of possible concepts and asked to express what is probably on the image only using the concepts provided. Abstracting from background knowledge, humans would probably first identify the objects on the image by parsing the annotation for entities (noun phrases). For every entity the evaluator would assess its semantic similarity to each of the provided concepts and select the most similar one.

SCM takes advantage of the Wordnet thesaurus to assess the similarity of a pair of concepts. Wordnet[2] groups English words into sets of synonyms called *synsets* and declares various semantic relations including hypernymy between the synsets in the form of a lexical semantic network. SCM expresses the desired set of classes as well as entities from text as Wordnet synsets. Wordnet similarity measure is then used to determine the similarity between an entity and each of the classes. Soft classification is thus achieved with an entity being classified to the class with which it has the highest similarity.

Although Wordnet is a comprehensive thesaurus containing approximately 146 thousand word–sense pairs for nouns (as of its version 3.0), it does not contain some uncommon words and most named entities. For the purpose of resolving entities not found in Wordnet, we introduced Targeted Hypernym Discovery (THD), which uses Wikipedia to find a hypernym to an entity.

The input for SCM is a free-text annotation and a custom-defined set of classes (Wordnet synsets) C_{tc}. NLP tools available in the GATE NLP Framework [10] are used to extract noun chunks from the annotation. Noun chunks are used similarly as in other approaches (e.g. [12]) to represent a semantic entity. SCM then maps each entity to a Wordnet synset, possibly with the help of THD, and computes the similarity between the synset and each of the classes in C_{tc}. The class with the highest similarity (confidence) is the prediction. However, the fuzzy character of the results facilitates classifier fusion [22].

2.1 Selection of Classification Concepts

SCM is an unsupervised classification algorithm, and no training is necessary for THD either. The classification performance is thus mainly affected by the similarity measure used and by the selection of a suitable set of synsets (classes) from the Wordnet thesaurus. As a consequence, if the same set of classes were used in SCM as in image classification and no regard were paid to the characteristics of the chosen similarity measure and to the position of the classes in Wordnet, the classifier performance would suffer.

Since SCM always produces a decision, the classes should ideally cover the whole universe of entities that may appear in the annotation, and not just the entities recognized by the image classifier. A theoretical option is to require some minimum similarity to classify an entity. If the similarity between the entity and the winning class was below a certain threshold, the entity would be classified as 'unknown'. The 'unknown' class would thus cover the part of the universe not covered by the image classifier. Since such thresholds proved difficult to find, the current framework requires that

[1]Such datasets only exist for standardized NLP tasks, such as named entity recognition.

[2]wordnet.princeton.edu

the 'unknown' is represented by multiple specific concepts (classes).

In this preliminary work, it is expected that classes for classification of entities from text C_{tc} are selected by a human expert. A possible discrepancy between the semantics of classes used by the image classifier and the semantics conveyed at the level of image annotations can thus be taken into account.

2.2 Mapping to Wordnet Synsets

SCM tries to map each noun phrase (entity) to a Wordnet synset according to the following experimentally-defined priorities: 1) noun phrase, 2) head noun, 3) hypernym for noun phrase and 4) hypernym for head noun. The match is successful if a Wordnet concept with the same string representation is found. If even the hypernym for head noun is not found, the system recursively extracts a more general hypernym mappable to Wordnet. The implementation of the hypernym discovery approach used is discussed in Section 3.

Consider the noun phrase 'Bucegi National Park', which has been extracted from an image annotation. Following the priorities outlined above, the system tries to look up the following 1) 'Bucegi National Park', which is not a Wordnet entry, and 2) 'park', which is a match since 'park' is a Wordnet entry. This result is correct, but can be improved by syntactical analysis of the noun phrase, which will allow more informed stripping of the modifiers (i.e. try 'national park' before 'park').

The general limitation of the current approach is posed by the fact that most queries have multiple matches in Wordnet. For example, there are six possible meanings (synsets) for the noun 'park' as given by Wordnet 3.0. The first three refer to a recreational area, the fourth one refers to the Scottish explorer Mungo Park, the fifth to 'parking lot' and the sixth to a gear position. The order of these entries is not random; Wordnet actually lists the most frequently used sense of the word first. The system uses the common baseline approach for word sense disambiguation [2] and only selects the most frequently used sense. Work in progress is focused on the development of a word sense disambiguation algorithm that would be able to identify such a combination of word senses that would maximize the overall similarity of the annotations in the collection.

2.3 Wordnet Similarity Measure

The system computes the similarity between the synset representing the entity and each of the custom-defined concepts (Wordnet synsets) in C_{tc}. There is a large body of work on Wordnet-based measures of semantic similarity [6]. Our system uses the *Lin* similarity measure. This measure has sound theoretical foundation stated in the Similarity Theorem [6] and is defined as

$$sim_L(c_1, c_2) = \frac{2 * log\, p(lso(c_1, c_2))}{log\, p(c_1) + log\, p(c_2)} \quad (1)$$

The function *lso* returns the lowest common subsumer from the hierarchy, and the value $-log(p(c))$ is called information content (IC). The value $p(c)$ denotes the probability of encountering an instance of concept c, which is estimated from frequencies from a large corpus. Our SCM implementation uses the Java Wordnet Similarity Library[3] (JWSL), which automatically derives the values of IC from the Wordnet structure by exploiting the hyponymy relations among synsets.

The experiment presented in subsection 5.2 evaluates the agreement between the class predicted by SCM based on Wordnet similarity and the human judgment.

[3]http://grid.deis.unical.it/similarity

3. TARGETED HYPERNYM DISCOVERY

The hypernym discovery approach proposed here is based on the application of hand-crafted lexico-syntactic patterns (Hearst patterns). Although lexico-syntactic patterns have been extensively studied since the seminal work [15] was published in 1992, most research have focused on the extraction of *all* word-hypernym pairs from the given generic free-text corpus. In contrast, the goal of Targeted Hypernym Discovery (THD) is not to find all hypernyms in the corpus but rather to find hypernyms for the current entity. Additional experiments presented here show that THD achieves a significantly higher accuracy than previous approaches to hypernym discovery. THD also has the advantage of requiring no training and can use up-to-date on-line resources to find hypernyms in real time. The THD algorithm proposed here is an updated and expanded version of the algorithm used in our earlier work [8]. The outline of the steps taken to find a hypernym for a given entity in our THD implementation is as follows:

1. Fetch documents from the corpus defining the entity
2. For each document:
   (a) Determine if suitable for further processing
   (b) Extract hypernyms matching the lexico-syntactic patterns
   (c) Return the most likely hypernym found

Performing all these steps requires to carry out multiple information retrieval and text processing tasks. For this purpose, our THD implementation uses the GATE NLP Framework [10]. Figure 2 sketches the NLP components and their interaction in THD. In the rest of this section, we substantiate the choice of Wikipedia as the corpus and explain the way it is interfaced and the documents are preprocessed. Finally, we focus on the application of lexico-syntactic patterns and provide an example of it.

3.1 Wikipedia as the Corpus

A gold-standard dataset for training and testing hypernym discovery algorithms is Wordnet (e.g. [25, 26]). Wordnet's structured nature and general coverage makes it a good choice for general disambiguation tasks.

The frequent occurrence of named entities in image annotations makes the use of most closed lexical resources including Wordnet unfeasible. This is documented in the study [25], which evaluated several hypernym discovery algorithms on a hand-labeled dataset where 60% of hypernyms were named entities. The performance of the best algorithm based on lexico-syntactic patterns significantly surpassed the best Wordnet-based classifier (F-Measure increase from 0.2339 to 0.3592).

The goal of THD is to improve the coverage of SCM by mapping entities that do not occur in Wordnet to Wordnet synsets through hypernyms extracted from a suitable

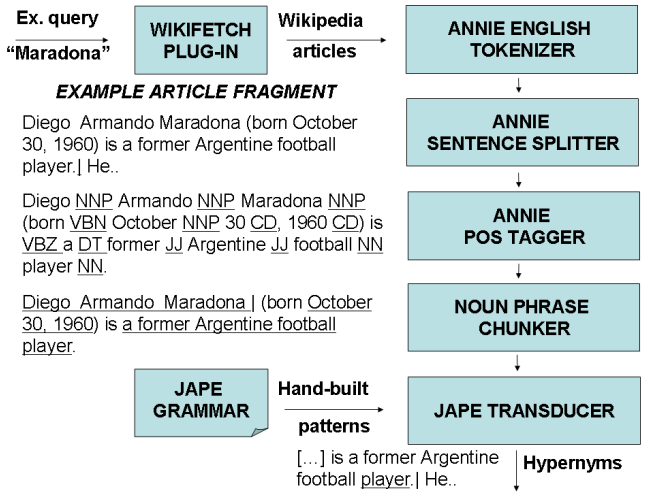

Figure 2: Targeted Hypernym Discovery

large free-text corpus. We opted for the fast growing, publicly available encyclopedia, Wikipedia, which contains more than two million articles (definitions) in English (as of June 2008).

Unlike [5] who combined web search and Wikipedia article titles and hyperlinks for extraction of instances of arbitrary relations or [27] who mainly use the Wikipedia category system for the purpose of ontology learning, we found the first section of Wikipedia articles as particularly suitable for hypernym discovery and use it as the sole source of information.

3.2 Interfacing Wikipedia

The selection of suitable articles for hypernym discovery is the main differentiator between the system presented here and other approaches in the literature. Our system interfaces with Wikipedia through a newly-designed Wikifetch plug-in for GATE. For a given entity (query for hypernym), Wikifetch executes an online search in English Wikipedia through the Wikipida Search API, which provides access to Wikipedia's Lucene-based fulltext search[4]. Articles are ranked, in addition to textual similarity, also based on the number of backlinks they recieve. This ensures that e.g. for the query "Gates" the first article in the search result list is an article on "Bill Gates", and not an article on some other person named Gates, which would have probably be produced on the basis of pure textual match. We assume that this feature ensures that the search results approximate the possible senses of the entity sorted in the descending order. Since in our current work we stick to this 'most frequent sense assumption', the articles are processed in the order of their appearance in the search results.

In many cases the article title is spelled differently or with a word missing or added as compared to the query. Extracting hypernyms from articles that only loosely match the query would deteriorate the performance of the system; it is therefore necessary to determine if the article is on the exact topic sought. In order to make this decision we compute a string similarity[5] between the article title and the original query. If this similarity is below an experimentally-set threshold, the article is excluded from further processing.

The system also performs diacritics stripping in order to improve the matching of non-English titles.

Since the capitalization of the query and the article topic should match, but Wikipedia capitalizes all the article headings, the article text is previewed to see if the topic of the article always appears in upper-case. This for example removes an article titled "Logical Gates" from the search result for query "Gates".

The full texts of n top ranked Wikipedia articles that passed the selection outlined above are obtained through the `Special:Export` interface of Wikipedia's MediaWiki engine[6], which puts less strain on Wikipedia's resources than crawling the Wikipedia would. Wikifetch strips away the wiki-markup, links, hidden text such as comments, information boxes etc., and returns the first section of the article.

3.3 Text Preprocessing

According to our experimental evaluation, the first section of each article provides a sufficient basis for THD since it contains a brief introduction of the topic of the article, often including the desired definition in the form of a Hearst pattern. Processing the remaining sections, in our experience, only increases the computation time and introduces noisy hypernyms.

The system uses the existing as well as newly-created GATE modules (see Figure 2) to perform text preprocessing. We use the modules available within the GATE reference information retrieval and extraction system ANNIE to perform text tokenization, sentence splitting and Brill-style part-of-speech (POS) tagging. Noun chunks are identified using the Ramshaw-and-Marcus chunker [21]. The system also performs customely implemented replacement of non-English characters by their ASCII fallback alternatives.

3.4 Pattern Matching

The NLP components that perform text preprocessing in the GATE framework append their output to the existing text in the form of annotations. Hearst patterns are usually matched in free text using regular expressions. Since here annotations are on the input, we use the JAPE engine [11], which evaluates regular expressions over annotations.

A Java Annotation Patterns Engine (JAPE) provides a finite-state transduction over annotations. Its input is a JAPE grammar (basically a set of rules) and a text to annotate. JAPE grammar rules consist of left- and right-hand side. On the left-hand side, there is a regular expression over existing annotations; annotation manipulation statements are the on the right-hand side. A JAPE grammar was already used to match Hearst patterns in [9]. However, their paper does not elaborate on the grammar used or on its performance in detail.

The JAPE grammar used in our research consists of several rules, which particularly differ in the way the entity for which a hypernym is sought is matched in the text. Each rule is assigned a priority, such as the strictest rule, which requires the entity to appear exactly as it is in the text, has

[4]http://www.mediawiki.org/wiki/Extension:Lucene-search

[5]Our system uses the Jaro-Winkler similarity, since this measure was specifically developed for matching named entities (people names) [29].

[6]http://www.mediawiki.org

the highest priority. The exactness of the required match decreases with the priority of the rule. If the text in the currently processed document matches multiple JAPE rules then the first match provided by the strictest rule is taken. If no rule fires then a next article provided by the Wikifetch plugin is processed. This is repeated until a hypernym is found or there are no more articles to process.

A sample pattern illustrating (using the text after the comment signs '//') the extraction of hypernym for the entity 'Maradona' as exemplified on Figure 2 follows:

```
//the rule matches patterns only within one sentence
Rule: HearstRuleExactMatch
Priority:1000

// rule-specific macro to match the query
(Query) // 'Maradona'
// matches any number of any tokens
({Token})* //'(born October 30, 1960)'
//followed by a form of "to be", here 'is'
{Token.string == "is"}|{Token.string == "are"}|
{Token.string == "were"}|{Token.string == "was"}
// followed by article, here 'a'
({Token.string == "a"}|{Token.string == "an"}|
{Token.string == "the"})
//followed by macro a defining allowed words
//preceding the actual hypernym for query
(NounChunkBody) // 'former Argentine football'
//hypernym can be only NN, NNS or NNP
(Head)  //'player'
:hearstPattern
--> //delimits LHS from RHS
//a new 'hearst' annotation is added to ''player'
//the string identified by the hearstPattern label
:hearstPattern.hearst = {rule = "ExactMatch"}
```

The NounChunkBody macro was determined experimentally and matches the following pattern: `Token? CD? JJ? JJ? NNP? NNP? VBN? JJ? JJ? NN? NN? NN? NN?`.

`Token?` matches any single token (word, comma etc.), `CD` matches a cardinal number, `JJ` an adjective, `NNP` a proper noun, `NN` a noun and `VBN` a verb. If this rule fires then it marks the hypernym, which can be a single noun (NN), a plural noun (NNS) or a proper noun (NNP), with the annotation 'hearst'.

When constructing this grammar we preferred speed to elegancy. Although the use of "+" and "*" operators would simplify and generalize the `NounChunkBody` macro, it would also have deteriorating impact on the processing speed, which we wanted to avoid. Other lexico-syntactic patterns identified by Hearst [15], e.g. the 'such as' pattern, were not considered, because they did not seem to provide a significant improvement from our observation.

THD has two outcomes: the (proper) noun annotated with the 'hearst' annotation ('player') and the noun chunk in which it is contained ('former Argentine football player'). This noun chunk can be in some cases identical with the noun, but ideally it should provide a less general hypernym for the query. This allows to map the original entity to a more specific concept of the thesaurus used.

3.5 Example

Consider the picture of a footballer scoring a goal, which is assigned the textual annotation "David Beckham hits the net again". This image is processed with the KAA image classifier trained on the sports domain. This classifier assigns the following labels (classes) to image segments: {*hockey player, football player, basketball player, swimmer, runner, sports equipment*}. In order to aid this classifier in its uneasy task, SCM/THD can be used to determine which of the classes probably appear in the image based on the textual annotation.

Semantic Concept Mapping first breaks the annotation into two entities, 'David Beckham' and 'net', and then attempts to map each of these entities to a Wordnet synset. Following the steps described in subsection 2.2, SCM tries to find an entry for 'David Beckham' in Wordnet, but there is no such entry. Since also the following try 'Beckham' fails, the system calls Targeted Hypernym Discovery to return a hypernym for 'David Beckham'. THD finds a Wikipedia entry entitled 'David Beckham', and using lexico-syntactic patterns it extracts the hypernym 'footballer'. SCM finds one synset that has the word 'footballer' attached, and maps 'David Beckham' to it. Then the system computes the similarity between this synset and the synsets representing each of the classes. The class 'football player' is correctly assigned the highest confidence; it has similarity 1 because it belongs to the same Wordnet synset as 'footballer'.

SCM then proceeds to the second entity, 'net'. THD is not used, because multiple synsets described with this word are found directly in Wordnet. SCM maps 'net' to its first, most frequently used, meaning (synset). Since this is 'the computer network' sense, the word gets misclassified. Luckily, the sports equipment meaning of the word is, nevertheless, assigned the highest similarity (0.60).

The result obtained with the on-line demo of our system is a feature vector:

```
//'David Beckham' mapped with THD to footballer
<football_player="1.0" sports_equipment="0.126"
hockey_player="0.720" runner="0.705"
swimmer="0.687" basketball_player="0.710">
//'net' found directly in Wordnet
<football_player="0.11" sports_equipment="0.60"
hockey_player="0.11" runner="0.34"
 swimmer="0.10" basketball_player="0.11">
```

The annotation is broken into two entities, 'David Beckham' and 'net'. 'David Beckham' is not found in Wordnet, THD is thus first used to map it to its hypernym 'footballer', which is mapped to a Wordnet synset, and then the similarity with each of the classes is computed. The entity 'net' is found directly in Wordnet, but 'horse' is incorrectly identified as the semantically closest class to it (similarity 0.64), the correct sense 'sports equipment', however, closely follows with similarity 0.63. This is due to the fact that the sports equipment meaning of 'net' is not the most frequent sense of the word.

In subsection 5.1 we present an experiment evaluating the performance of THD on a real dataset.

4. TEXT-ENHANCED KAA

There are multiple promising scenarios where merging the results of analysis of image content and the accompanying free-text annotation would be beneficial. SCM/THD breaks textual annotations into entities and classifies them into a custom-defined set of classes. This result can be used in the classical image classification task, where the whole image is

assigned one label. Since the outcome of SCM are multiple named entities, it is natural to fuse these results with those of Knowledge Assisted Image Analysis (KAA), which also detects and classifies multiple objects (i.e. segments) on an image.

In the following we will briefly outline the KAA implementation used [20], and present a tentative approach fusing the SCM and KAA classification results. We emphasize that the purpose of this task is to illustrate how the presented approach to text analysis can contribute to image classification. We will focus on more effective approaches to the fusion of these classifiers in our future work.

4.1 Knowledge-Assisted Analysis

The objective of KAA is to label image regions from a predefined set of semantic concepts C_{ic} = *rock, sky, person, ...* Each region is assigned one concept from this set. For classification we use a self-organizing map (SOM), whose performance is improved by particle swarm optimization (PSO). A detailed discussion of the algorithm is presented in our previous work [7]. Prior to classification, the images are segmented into image regions using the RSST algorithm [1], and for each image region, MPEG-7 low-level visual features are extracted [18]. In our earlier work [19] we developed a framework for evaluating the performance of multiple classification methods: support vector machines (SVM); genetic algorithms (GA) in combination with SVM; SOM alone; SOM+PSO.

From the careful observation of the results we concluded that the use of optimization methods (such as GA or PSO) in combination with a more traditional classifier (SVM and SOM, respectively) generally leads to increased classification accuracy compared to using the latter classifiers alone [19]. Furthermore, the use of an increased number of images for training the classifiers is generally beneficial, highlighting the need for the availability of large annotated media sets for appropriately training any classification method. However, even in the absence of a rich training set we showed that meaningful classification results can be produced; the PSO classification utilized in the KAA implementation used in this paper is shown to be particularly suitable in this case. Experimental evaluation of this KAA implementation is presented in Experiment 3.

4.2 Fusing the Results of SCM and KAA

The analysis of image annotations cannot in the general case provide information on objects present in the image comparable in terms of completeness to the analysis of the image content. However, entities appearing in image annotation are likely to occur in the image [12]. This section presents an example approach to merging the classification results of KAA and SCM with the goal of determining the most important concept present on the image.

Our approach relies on the assumption that the most important concept should be present on at least one image segment and at the same time mentioned in the image annotation.

Each image is assigned a distinct set of classes $T, T \subseteq C_{tc}$, which were assigned by SCM to entities in the text, and a distinct set of classes $V, V \subseteq C_{ic}$, which were assigned to segments of the image by KAA. In order to compute the desired intersection of the classification results, we express the results of SCM in terms of classes used by KAA using the following transformation:

$$T \xrightarrow{f} T_f, T_f \subseteq C_{ic}$$

The projection $f : C_{tc} \rightarrow C_{ic}$ is defined by a human expert.

Concepts $c \in C_{tc}$ that do not have their counterpart in C_{ic} are projected to the empty set, $c \rightarrow \emptyset$. A similar solution is proposed by [12] who mark 25 manually selected seed concepts as either visual or non-visual; the visualness of a given concept is determined using the Wordnet similarity between the entity and the seeds. Entities with visualness below a certain threshold are discarded.

The intersection of the classification results from the textual and image analysis $COMB = T_f \cap V$ provides the basis for the selection of the image class. If $COMB$ contains multiple concepts or the intersection is empty then we prefer the concept selected by KAA, since it has more complete information.

In Experiment 4 we evaluate this simple approach to combining the outcomes of SCM and KAA. Future work that will address the shortcomings of the current approach, particularly the need for a different set of concepts for SCM, is discussed in Section 7.

5. EVALUATION

In the first three experiments we individually evaluate our implementations of THD, SCM and KAA. In Experiment 4 we combine the results of the systems to show how the analysis of text can contribute to image classification.

There are no standard datasets available for the tasks performed here. The closest available dataset for the evaluation of THD/SCM is perhaps the one used in the ACM KDD CUP 2005 for Query Categorization. This dataset was not used since search engine queries exhibit significantly different linguistic properties than free-text image annotations and there are no accompanying images to demonstrate the fusion. The Corel and Washington DC data sets do not contain free-text annotations, hence, similarly as in related research dealing with free-text annotations [12, 4, 28], we used a proprietary set of annotated images.

The datasets used in most experiments are comparable in size with some of the related work. The number of hypernyms in the human-annotated test set of [25] was 131, while in our Experiment 1 there were 98; [12] mentions processing 100 image annotations vs. 105 in our Experiment 2; and in [14] 50 images were classified compared to our 489 segmented regions in Experiment 3. Our weakest dataset is for Experiment 4, where we use 92 images.

Implementations of SCM and THD used in the experiments are available as an online demo[7].

5.1 Exp 1: Targeted Hypernym Discovery

Our system differs in two fundamental ways from existing approaches in hypernym discovery: a) it performs *targeted* hypernym discovery, only selecting the most suitable hypernym for each query, and b) it is focused on discovery of hypernyms for named entities and uncommon words. These factors influenced the choice of the evaluation procedure and the test datasets. This experiment only aims at evaluating the hypernym extraction from Wikipedia articles, assuming

[7] http://nb.vse.cz/~klit01/hypernym_discovery/

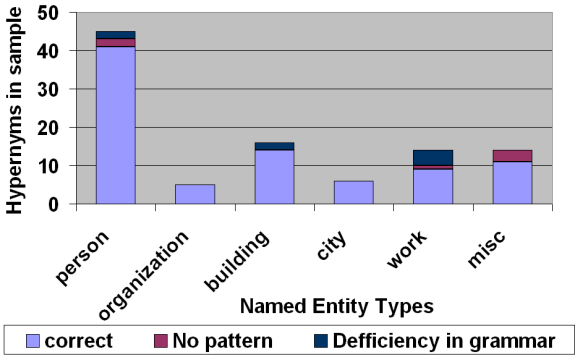

Figure 3: THD accuracy per named entity type

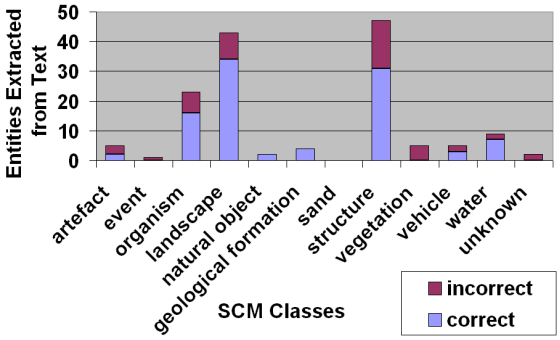

Figure 4: SCM classification result

that an article defining the given entity is available. Note that THD is evaluated on a more comprehensive task in Experiment 2.

We randomly selected 100 articles describing named entities from Wikipedia using the 'random article' link. Article titles were used as queries for hypernym and the articles as the corpus.

THD executed on this test set correctly discovered hypernyms for 86 Wikipedia article titles. The system failed to extract the correct hypernym from 14 articles: a Hearst-like pattern was not present in 8 articles, out of which 2 did not even contain any hypernym, and in 6 cases a Hearst-like pattern was present but was not matched by the extraction grammar. A detailed analysis of the results depending on entity type is depicted on Figure 3. An encouraging result from the point of view of integration of THD with SCM is that all the discovered hypernyms were mappable to Wordnet with a disambiguation accuracy of 87% for the most frequent sense synset.

The overall accuracy[8] achieved in the experiment was 88%. We consider this as a very good result but we cannot provide a benchmark, since we perform *targeted* hypernym discovery, while most related approaches including [25] try to discover all hypernym pairs from the corpus. However, we believe that the results show that THD is an effective approach for resolving named entities.

5.2 Exp 2: Semantic Concept Mapping

The goal of this experiment is to evaluate how well the system presented here is able to map entities extracted from a specialized image collection to Wordnet concepts as compared to human judgment. We used a collection of 1276 images taken by a professional photographer during trips to Albania and Romania as the test set. These images have short textual annotations consisting of 1 to 10 words saved in the EXIF data. Out of the available annotations we extracted 105 images with unique annotations.

We decided to perform the evaluation at the entity level, since the number of entities per image varied, which would make image-level comparison difficult. We used the following set of eleven Wordnet concepts as the set of classes:[9] C_{tc} ={*natural object, artifact, event, vehicle, sand, geological formation, structure, organism, water, vegetation, landscape*}. This selection reflected the needs of the semantic concept detection as introduced in Experiment 4.

Two annotators were asked to select the semantically closest concept for each of the entities. The annotators were allowed to use Wikipedia. Even then, for two entities, *Jetee du Dragon* and *Syri i Kalter*, the annotators were both unable to assign a label.

The annotations were first broken by SCM into 196 entities and mapped to Wordnet synsets. The resulting accuracy was 85% (167 correct classifications). Out of the 196 entities, 95 (49%) were named entities not present in Wordnet. THD correctly found a hypernym for 71 of these named entities (accuracy 75%). THD thus accounted for 83% of the error. The remaining 17% error was due to the most-frequent-sense assumption, which caused the selection of a wrong Wordnet synset, and to incorrect noun phrase chunking and entity detecion.

A similar task was perfomed by [12], who classified all entities appearing in 100 annotations assigned to images from Yahoo! News. Their annotations were longer (15 entities per image on average) and comprised several sentences. Using a combination of a Named Entity Recognition (NER) system and Word Sense Disambiguation (WSD) package, the authors achieved an accuracy of 75.97% when classifying entities to Wordnet synsets. The erroneous entity detection accounted for 32.32% of the error, 60.56% was caused by the WSD system and 8.12% by the NER package. It is difficult to make a comparison between the sources of error, since [12] did not give the number of named entities in their dataset.

The experiment of [12] finished with disambiguating the extracted entities to Wordnet synsets, since they did not attempt to align the results of text analysis with those of the visual analysis. In contrast, we used SCM to map the disambiguated entities to an arbitrary set of Wordnet concepts C_{tc} in order to support such alignment in Experiment 4. Our system achieved an overall accuracy of 52% on this task. The achievable maximum given by inter-annotator agreement was 80%. This compares favourably with the most-frequent-sense baseline of 24%, which assigns all the noun chunks to the most common concept (here 'structure').

[8]We do not give precision, recall and F-measure for our result since it is not clear whether to count an incorrect hypernym as false positive (the system gave a wrong answer) or false negative (there was a good answer).

[9]Each concept was represented by its most frequent sense.

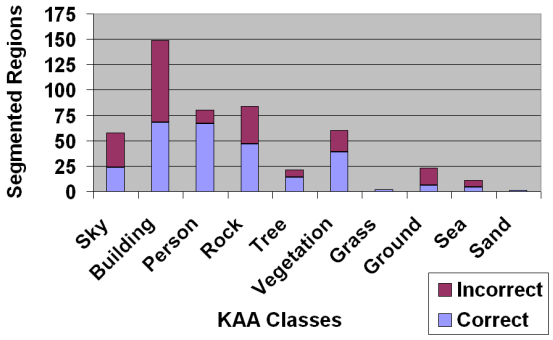

Figure 5: KAA classification results

5.3 Exp 3: Knowledge-Assisted Analysis

The evaluation of KAA was performed on 489 labeled segmented regions from the same 105 images.

Region classification was performed for the following 10 concepts C_{ic} ={ *Sand, Sea, Vegetation, Person, Sky, Rock, Tree, Grass, Ground, Building*}. The classifier for these concepts was trained based on a dataset of 50 images from our previous work [19].

The methodology used for evaluation of classification results was the same as in the TRECVid high-level feature detection task. The classifier was trained without the knowledge of the test set. A region where the result was the same as the human annotation was considered as correctly classified, while the opposite case was counted as error.

The results of the analysis are depicted in Figure 5. The overall accuracy of the system was 56%. Not considering the marginally present *Grass* and *Sand* concepts, the system was most successful in classifying the *Person* concept with 84% accuracy and the *Tree* concept with 67% accuracy.

5.4 Exp 4: Combining Text with Images

The goal of this experiment was to verify that combining the results of SCM and KAA can be beneficial for image classification. We used the same test of 105 images as in Experiments 2 and 3. The ground truth was provided by a human who annotated each image from the ground truth with one concept from C_{ic}; 13 images where the annotator could not decide were discarded.

In the experiment we attempted to improve the baseline classification performance by fusing the results of KAA with SCM as suggested in subsection 4.2. Table 1 defines the projection $f : C_{tc} \rightarrow C_{ic}$, which is necessary to map the results of SCM to the same set of semantic concepts as in KAA (and in the ground truth).

It should be noted that C_{tc} was selected with regard to the character of the dataset and to the possibility of manually defining the mapping to C_{ic}. Since from past experience we knew that our KAA classifier tends to classify different kinds of standing structures as buildings, *structure* was included to C_{tc} and this was reflected in f. For similar reason, *natural object* is mapped to *Tree* and *landscape* to *Vegetation*. The C_{ic} concepts *Sky*, *Grass* and *Ground* were not included in C_{tc} since these are common concepts rarely explicitly mentioned in annotations. The concepts *Vehicle*, *Event* and *Artifact* were only included in C_{ic} in order to capture entities that are not detected by KAA.

sand→sand	structure→building	vegetation→veg.
water→sea	landscape→vegetation	artifact→∅
geo. for.→rock	organism→person	event→∅
nat. obj.→tree	vehicle→∅	→

Table 1: Projection $f : C_{tc} \rightarrow C_{ic}$

We provide two baselines, one for the textual and one for the visual classifier. For the SCM (text-only) baseline, the results of SCM were projected using f and the concept with the highest confidence was selected as the image label. For the KAA (image-content only) baseline, the label for the whole image was provided by the class associated with the segment with the highest region importance, which was computed based on the percentage ratio between the area of the segmented region and the whole image. In our test-set the SCM baseline resulted in an accuracy of 27% and the KAA baseline in an accuracy of 42%.

The fusion of the SCM and KAA classifiers resulted in an accuracy of 55%. This is a 49% relative improvement over the text-only baseline and 31% relative improvement over the image-content only baseline. Remarkably, the analysis of the results showed that complementing the KAA result with that of SCM improved the classification performance to the person class by 20% compared to the KAA baseline. This would underpin the effectiveness of the system in resolving named entities, but further experiments on a larger and more varied dataset are needed to confirm this result.

Inspection of the result showed that the manually provided mapping f and the selection of concepts C_{tc} were not ideal. Further research in this area is necessary to improve the classification results and to eliminate the need for human assistance.

6. RELATED RESEARCH

This paper refers to techniques from several research areas, particularly from hypernym discovery, named entity recognition, word sense disambiguation and image classification. We tried to reference the most relevant works from each of these disciplines within the respective parts of this paper. In this section we only focus on works combining textual annotations with image analysis.

There is relatively small number of papers that report free-text image annotations as an aid for image classification. According to [12] the earliest system was NameIt! [23], which associated names with faces in news video using the analysis of video captions and extraction of named entities from transcripts. The performance of named entity extraction was poor (13% precision), but the overall results were promising (33% accuracy of name-to-face retrieval). A more recent approach to a similar task, presented in [4], aready used a NER system to improve the accuracy of extraction of named entities.

The paper [12] determines if entities extracted from an image annotation appear in the image. They detect and classify all entities (not just persons) but do not work with visual information. This research can be considered as the closest to our work in that noun chunks are also extracted from text and mapped to Wordnet synsets. The authors use Wordnet to determine whether the entity is visual, but do not perform mapping to a custom-defined set of classes. The

recognition of person names is improved through a dictionary of names extracted from Wikipedia.

It should be noted that SCM/THD has the advantage that, in principle, all entities in text can be mapped to a custom-defined set of concepts. In contrast, NER systems only categorize named entities to several predefined classes (typically 'organisation', 'person', 'location', 'miscellaneous' [13]). Retraining NER systems for a different set of classes is expensive as a large training set is necessary.

There are also multiple more distantly related approaches, especially from the area of information retrieval. For example [28] uses LSI to represent information coming from both the image and the textual analysis in one semantic space. The image annotations are represented by full-term vectors; no NLP is performed. The authors note that LSI as a statistical technique is less useful for named entities since these often occur infrequently in the corpus.

Of interest is also the work of [14], which combines the textual content with image features to classify images into four categories based on on the text surrounding the images on web-pages. No NLP or NER was performed, and the use of textual content resulted in marginal improvement in classification to categories for which named entities were important. This can be accounted to problems with statistical processing of named entities, as also marked by [28].

7. DISCUSSION AND FUTURE WORK

The review of related research presented in Section 6 has shown that most approaches that exploit textual information for image classification either ignore uncommon words or use a NER system or its variation. The disadvantage of most NER techniques is that they are not flexible enough to accommodate the variable needs of image classification, since large labeled corpora are needed for their training. To the best of our knowledge, our framework based on Semantic Concept Mapping and Targeted Hypernym Discovery constitutes one of the first attempts to harness the information about image content contained in named entities and uncommon words appearing in free-text annotations while not constraining the set of classification categories.

Experiments 1 and 2 showed that Targeted Hypernym Discovery is an effective tool for mapping uncommon entities to Wordnet. The current implementation of THD heavily relies on the first-sense assumption: the system processes articles in the order returned by Wikipedia search, mapping the first hypernym found to its first Wordnet synset. Relaxing this assumption could perhaps improve the THD performance. It is however the Semantic Concept Mapping that should be the primary focus of further work since it was responsible for the largest portion of error in the experiments.

The fact that hypernyms extracted from Wikipedia are too fine-grained for classification to general categories used in the NER task has been already noticed in [17] and the same apparently applies to the set of classes used in our experiments. Another problem is highlighted in Section 2.1, which recommends to use a different set of classes in SCM than in the image classifier, since the results of the similarity function used are very sensitive to the position of the class in the Wordnet hierarchy.

We suggest that future work should focus on replacing the Lin similarity, which we used in the experiments, with a more robust measure. Recent results achieved in the NLP community [24] demonstrated the superior performance of combining similarity measures based on semantic networks (e.g. Lin or JCn) with measures that use textual concept definitions (such as the Lesk similarity).

The original Lesk similarity computes the overlap between the Wordnet definitions of the compared concepts [3]. Inspired by the Extended Lesk similarity [3], we suggest to represent each concept (i.e. entity or class) with multiple Wikipedia articles related to it. We hope that the larger amounts of text thus provided by Wikipedia might provide results less dependent on the exact position of the compared entities in Wornet. A less volatile measure would also allow to define a minimum similarity threshold under which an entity would be classified as unknown. This should help remove the need to specify different classes for SCM than are used for image classification.

The soft output of SCM and the fact that it makes different errors than the image classifier, as shown in Experiments 2 and 3, facilitates the application of classifier fusion [22]. Experiment 4 demonstrated the positive impact of combining SCM with KAA, a region-based image classifier, using a simple classifier fusion algorithm. Investigation of effective approaches to fusion of SCM with a general image classifier was left for future research.

8. CONCLUSIONS

The paper presented an approach to utilizing textual annotations to complement image classification. Our contribution is two-fold.

First, our system uses Semantic Concept Mapping to express entities occurring in free-text image annotations in terms of custom-defined Wordnet synsets, and Targeted Hypernym Discovery to map named entities and uncommon words occurring in the text to Wordnet by extracting hypernyms from Wikipedia using lexico-syntactic patterns.

Second, we experimentally demonstrated the positive impact of complementing content-based image classification with entities extracted from free-text image annotations.

Further research will particularly focus on improving the accuracy of our SCM system by employing word sense disambiguation algorithms and on evaluating the benefits of the proposed approach for multi-class classification of images.

9. ACKNOWLEDGMENTS

The research leading to this paper has been partially supported by the European Commission under the IST research network of excellence K-SPACE of the 6th Framework programme, the work of Tomas Kliegr is also supported by grant GACR 201/08/0802 of the Czech Grant Agency. The authors would also like to thank Thanos Athanasiadis and his group for their contribution in developing the KAA system.

10. REFERENCES

[1] T. Adamek, N. O Connor, and N. Murphy. Region-based segmentation of images using syntactic visual features. *WIAMIS'05: Workshop on Image Analysis for Multimedia Interactive Services, Montreux, Switzerland*, 2005.

[2] E. Agirre. *Word Sense Disambiguation: Algorithms and applications*. Springer, 2007.

[3] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, 2003.

[4] T. L. Berg, A. C. Berg, J. Edwards, and D. Forsyth. Who's in the picture? In *Neural Information Processing Systems Conference*, 2004.

[5] S. Blohm and P. Cimiano. Using the web to reduce data sparseness in pattern-based information extraction. In *PKDD*, volume 4702 of *Lecture Notes in Computer Science*, pages 18–29. Springer, 2007.

[6] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[7] K. Chandramouli. Image classification using self organising feature maps and particle swarm optimisation. In *Doctoral Consortium of SMAP'07: Proceedings of 2nd International Workshop on Semantic Media Adaptation and Personalization*, pages 212–216, 2007.

[8] K. Chandramouli, T. Kliegr, J. Nemrava, V. Svátek, and E. Isquierdo. Query refinement and user relevance feedback for contextualized image retrieval. In *VIE 08: Proceedings of the 5th International Conference on Visual Information Engineering*, 2008. To appear.

[9] P. Cimiano and J. Voelker. Text2onto - a framework for ontology learning and data-driven change discovery. In *NLDB'05: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems*, 2005.

[10] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. A framework and graphical development environment for robust NLP tools and applications. In *ACL 2002*, 2002.

[11] H. Cunningham, D. Maynard, and V. Tablan. JAPE - a Java Annotation Patterns Engine (Second edition), Department of Computer Science, University of Sheffield, 2000. Technical report.

[12] K. Deschacht and M.-F. Moens. Text analysis for automatic image annotation. In *ACL'05: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 1000–1007, Prague, Czech Republic, June 2007. ACL.

[13] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA, 2005. ACL.

[14] T. Gevers, F. Aldershoff, and A. W. Smeulders. Classification of images on the internet by visual and textual information. In *Proceedings of SPIE Conference on Internet Imaging*, pages 16–27, Dec. 1999.

[15] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics*, pages 539–545, 1992.

[16] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *ECML-98: Proceedings of the 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[17] J. Kazama and K. Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *EMNLP-CoNLL'07: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.

[18] B. S. Manjunath, J.-R. Ohm, V. V. Vinod, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on MPEG - 7*, 11(6):703–715, June 2001.

[19] G. T. Papadopoulos, K. Chandramouli, V. Mezaris, I. Kompatsiaris, E. Izquierdo, and M. Strintzis. A comparative study of classification techniques for knowledge-assisted image analysis. In *WIAMIS'08: Proceedings of 9th International Workshop on Image Analysis for Multimedia Interactive Services*, 2008.

[20] G. T. Papadopoulos, V. Mezaris, I. Kompatsiaris, and M. G. Strintzis. Combining Global and Local Information for Knowledge-Assisted Image Analysis and Classification. *EURASIP Journal on Advances in Signal Processing*, 2007:1–15, 2007.

[21] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *ACL Third Workshop on Very Large Corpora*, pages 82–94, 1995.

[22] D. Ruta and B. Gabrys. An overview of classifier fusion methods. *Computing and Information Systems*, 7, 2000.

[23] S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and detecting faces in news videos. *IEEE MultiMedia*, 6(1):22–35, – 1999.

[24] R. Sinha and R. Mihalcea. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 363–369, Washington, DC, USA, 2007. IEEE Computer Society.

[25] R. Snow, D. Jurafsky, and A. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*, number 17, pages 1297–1304, Cambridge, MA, 2005. MIT Press.

[26] R. Snow, D. Jurafsky, and A. Ng. Semantic taxonomy induction from heterogenous evidence. In *COLING/ACL 06*, pages 801–808, Sydney, Australia, 2006. ACM.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW 2007: 16th international World Wide Web conference*, New York, NY, USA, 2007. ACM Press.

[28] T. Westerveld. Image retrieval: Content versus context. In *Content-Based Multimedia Information Access, RIAO*, 2000.

[29] W. E. Winkler and Y. Thibaudeau. An application of the fellegi-sunter model of record linkage to the 1990 u.s. decennial census. Technical report, U.S. Bureau of the Census, Washington, D.C., 1991. Statistical Research Report Series RR91/09.