SEVENTH FRAMEWORK PROGRAMME

Deliverable number:	D1.1
Deliverable Title:	Text corpora and image databases
Type (Internal, Restricted, Public):	PU
Authors:	Daiva Vitkute-Adzgauskiene, Irena Markievicz, Tomas Krilavicius, Minija Tamosiunaite, Tomas Kulvicius, Leon Bodenhagen, Hagen Langer
Contributing Partners:	AAU, UoB, SDU, VMU, UGOE

Project acronym:	ACAT
Project Type:	STREP
Project Title:	Learning and Execution of Action Categories
Contract Number:	600578
Starting Date:	01-03-2012
Ending Date:	28-02-2015

Contractual Date of Delivery to the EC:	28-02-2014
Actual Date of Delivery to the EC:	28-02-2014

Content

1. Executive summary

This document presents a summary and basic statistical data for the formed task-specific corpora and image databases. For both corpora and image databases we first give a short description of the procedures used for data acquisition, cleaning and storing, as well as the description of data sources used. Then we describe the structure of the accumulated corpora and image databases, including metadata structure used for annotation purposes. For text corpora, we additionally provide the results of corpus content analysis, showing the representativeness of corpora texts, covering the distribution of distinct action verbs and distinct terms denoting action background elements.

2. Introduction

This document presents a summary and basic statistical data for the formed task-specific corpora and image databases. It is important to accumulate extensive text and image data in order to provide qualitative input for other ACAT activities, first of all for process memory formation. Process memory formation is a data-driven process where topic related text and image material is analyzed in order to extract action verbs and verb-associated objects thus forming a backbone for Action Category formation. The quality of the Action Category back-bone is directly dependent on the number of distinct action verbs and action category objects, that can be extracted from the accumulated data, and, for this reason, basic statistical data for the accumulated corpora and image databases is provided, in line with the description of the procedures and sources used for data acquisition and cleaning.

The goal of this document is to present the current status of accumulated corpora and image databases for two ACAT project scenarios – CHEMLAB and IASSES. The thorough description of two demonstrator scenarios and related instruction sheets is presented in D5.1. Only a brief overview of these scenarios is presented in this document in order to show the content-requirements for the accumulated corpora and image databases.

3. Task-specific corpora and image databases

In ACAT, both corpora and image data are accumulated for two main scenarios – CHEMLAB and IASSES. Procedures, used for data accumulation are similar for both scenarios, while the main difference lies in the sources used. CHEMLAB and IASSES scenarios are thoroughly described in D5.1, and only a very brief description is given in this section in order to be able to show the conformity of the accumulated corpus and image data.

The selected CHEMLAB scenario is the process of DNA extraction from a sample. The process involves the handling of liquids (pouring, decanting, etc.) and usage of standard laboratory equipment such as jars of different size and shape, filter cartridges, and a centrifuge. In order to be successful the process has to be executed under the required constraints (temperature, time schedule, etc.) stated in the respective lab protocol.

The IASSES scenario focuses on manufacturing tasks from the production of rotors for submersible pumps at the SQ-factory at the Danish company Grundfos.

Both the accumulated corpora and image databases are stored on a subversion (SVN) server, dedicated to the ACAT project (URL http://kleinas.vdu.lt/svn/ACAT-416859).

3.1. Formation of task-specific corpora

The main points, characterizing the formed corpora for CHEMLAB and IASSES scenarios of ACAT, are:

1) Procedures for corpus data acquisition and cleaning, data sources.
2) Corpus structure, including metadata structure.
3) Summary and statistics for the accumulated corpora.

3.1.1. Procedures and data sources

Procedures for corpus data acquisition and cleaning and their sequence are presented in Figure 1.

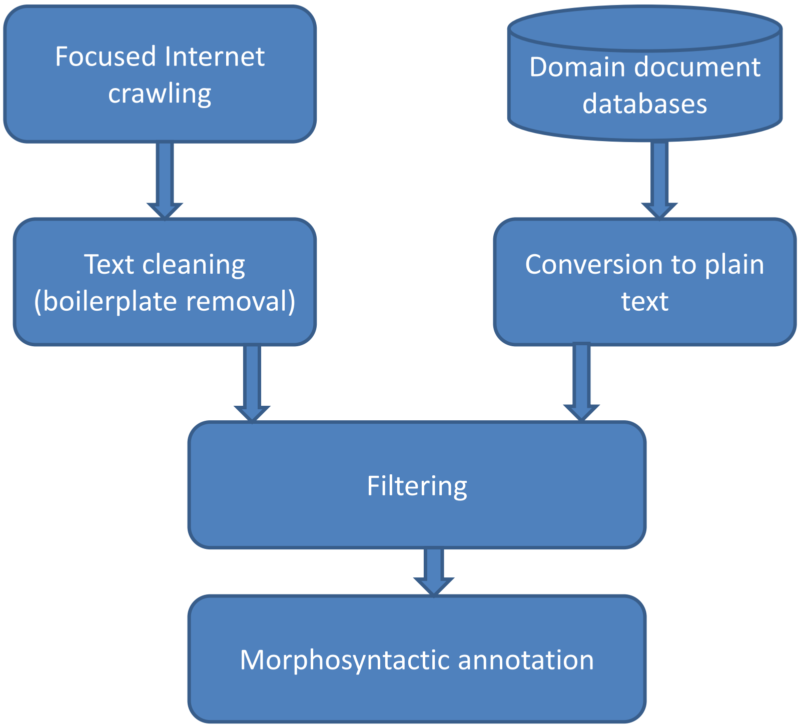

Focused Internet crawling
Domain document databases
Text cleaning (boilerplate removal)
Conversion to plain text
Filtering
Morphosyntactic annotation

Figure 1*: Procedures for corpus acquisition and cleaning*

Both crawling of freely available Internet resources and use of specific domain-focused document databases are employed for corpus data acquisition. Crawling is executed by applying a focused crawler, using domain-specific keyword lists, accumulated by applying pre-analysis of domain specific texts and expert input, and an URL list at its input.

Crawling was executed using the *Apache Nutch* scalable open source crawler (http://nutch.apache.org/). Figure 2 presents the crawling scheme. *CrawlDB* database maintains info on all known URL: fetch schedule, fetch status, page signature and metadata. *LinkDB* database includes list of source urls. Segments database keeps crawled web page context, it‘s plain text and outlink. *Injector* module is responsible for adding new urls. *Generator* and *Fetcher* prepare fetchlist and download content. Crawled data is preprocessed by *Parser* module. It cleans fetched content from html tags and additional information. Each crawled website can be periodically updated with *Updater* module. *Link inverter* allows to crawl data from the inner urls of already fetched websites. Finally, crawled data is indexed (*Indexer* module) and sent to *Sorl Indexer* to remove duplicates.

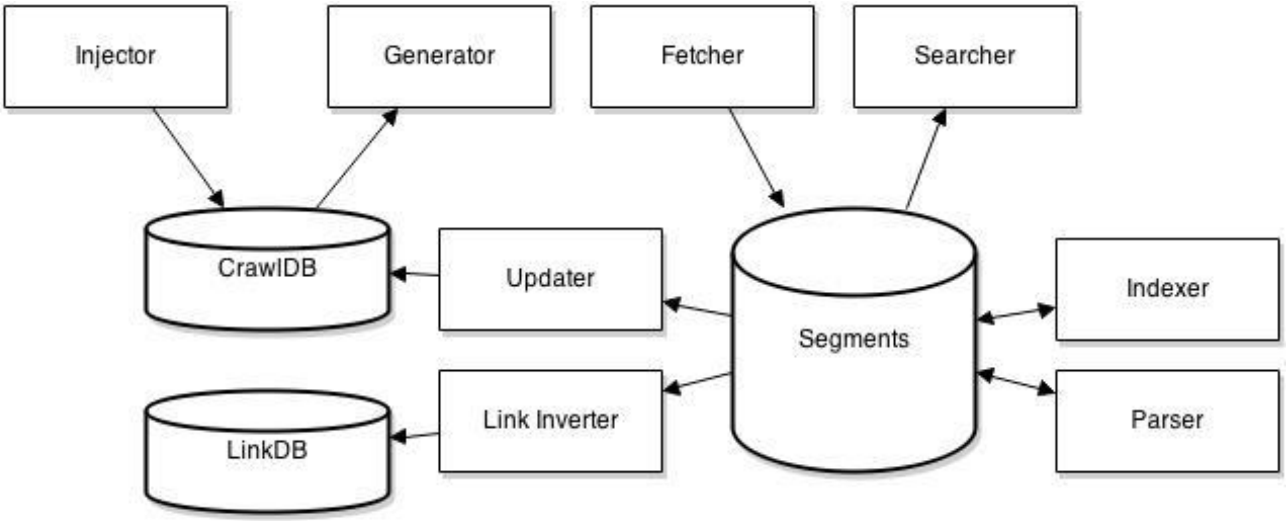


Figure 2*: Crawling scheme using the Apache Nutch crawler (source: https://**nutch.apache**.org/)*

Domain-specific document databases, used for corpus text collection, include available industrial databases with task specific manuals, training materials and scientific papers.

In the first processing layer, corpus data is cleaned using boilerplate removal schemes, i.e. detecting and removing the surplus "clutter" (HTML tags, templates) around the main textual content. PDF documents are converted to the plain-text format, using *PDFBox* tools (http://pdfbox.apache.org/). *PDFBox* is an open source Java PDF library for working with PDF documents. The *PDFBox* library allows creation of new PDF documents, manipulation of existing documents and the ability to extract content from documents. It is a common *Apache Nutch* plugin for indexing PDF documents. There is also possibility to integrate *PDFBox* with *Apache UIMA* for document unstructured content analysis.

Plain-text, obtained as the result of the first processing layer, is the supplied to the second processing layer, dedicated to additional text filtering. In this layer, keyword lists and stop-word lists are used to filter out texts, which are irrelevant or weakly linked to the domain.

Finally, morphological annotation of the corpus texts is accomplished using Stanford tools for morphological analysis (http://nlp.stanford.edu/software/). This annotation level is obligatory in order to be able to identify action verbs and action object categories in the text. The final result of this processing level is an XML document for each corpus text.

For the CHEMLAB scenario, Internet crawling was used as the main source of information. The process was executed in iterations, with the resulting lists of extracted keywords from one iteration used as focusing info for the next iteration. Also, documents with tutorials and training material on chemical and biotechnological experiments were used, as well as scientific documents from the PUBMED database (http://www.ncbi.nlm.nih.gov/pubmed).

For the IASSES scenario, the main sources of information are collections of different manual documents in PDF format.

3.1.2. Corpus structure

The size of the accumulated **CHEMLAB** corpus is **8919087** running words. It is structured in the following way:

- 33,84% (3018220 running words) - general chemistry texts, crawled from internet, mainly tutorials for chemical experiments;
- 41.51% (3702313 running words) - biochemistry and biotechnology texts, crawled from the Internet;
- 24.65% (2198554 running words) - biochemistry and biotechnology texts from PUBMED electronic library.

Further analysis of accumulated CHEMLAB texts is done separately for the general chemistry part (marked as CHEMLAB) and biochemistry and biotechnology part (marked as BIOCHEM).

The size of the accumulated **IASSES** corpus is **3563775** running words. It consists of manuals, assembly instructions and descriptions, crawled from the Internet and obtained from project partner document libraries.

Both CHEMLAB and IASSES corpora is available in two formats:

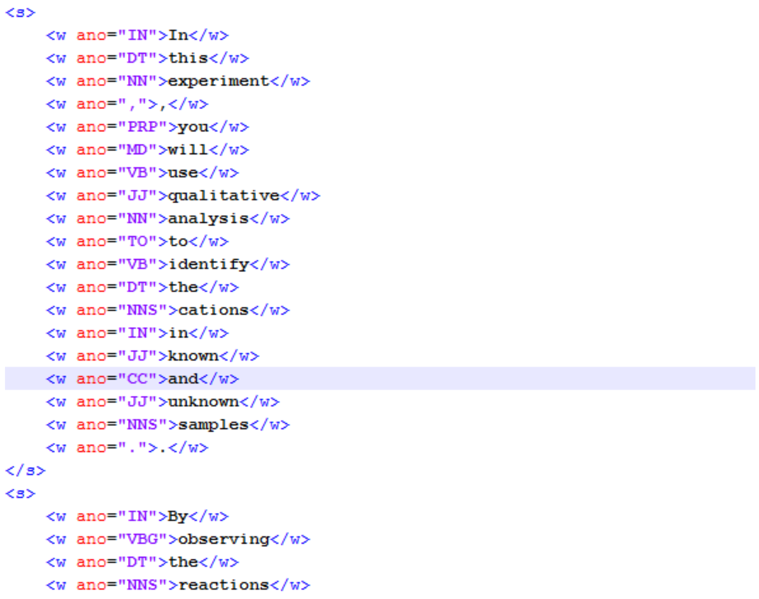
1) Plain text format;
2) XML format with morphological tags added.

The morphological tags for CHEMLAB and IASSES corpora are formed, using Stanford annotation tools and POS (Part-Of-Speech Tagging Guidelines), designed for the Pen Treebank Tagging Project (http://repository.upenn.edu/cis reports/570).

The list of tags used for morphological annotation is presented in Table 1. Figure 3 presents an excerpt from a morphologically tagged CHEMLAB corpus. Annotations for the IASSES follow the same metadata structure.

Coordinating conjunction	CC
Cardinal number	CD
Determiner	DT
Existential there	EX
Foreign word	FW
Preposition or subordinating conjunction	IN
Adjective	JJ
Adjective, comparative	JJR
Adjective, superlative	JJS
List item marker	LS
Modal	MD
Noun, singular or mass	NN
Noun, plural	NNS
Proper noun, singular	NP
Proper noun, plural	NPS
Predeterminer	PDT
Possessive ending	POS
Personal pronoun	PP
Possessive pronoun	PP$
Adverb	RB
Adverb, comparative	RBR
Adverb, superlative	RBS
Particle	RP
Symbol	SYM
to	TO
Interjection	UH
Verb, base form	VB
Verb, past tense	VBD
Verb, gerund or present participle	VBG
Verb, past ~articiple	VBN
Verb, non-3rd person singular present	VBP
Verb, 3rd person singular present	VBZ
Wh-determiner	WDT
Wh-pronoun	WP
Possessive wh-pronoun	WP$
Wh-adverb	WRB

Table 1*: Tags used for morphological annotation of ACAT corpus data (source:* http://repository.upenn.edu/cis reports/570).



```
<s>
    <w ano="IN">In</w>
    <w ano="DT">this</w>
    <w ano="NN">experiment</w>
    <w ano=",">,</w>
    <w ano="PRP">you</w>
    <w ano="MD">will</w>
    <w ano="VB">use</w>
    <w ano="JJ">qualitative</w>
    <w ano="NN">analysis</w>
    <w ano="TO">to</w>
    <w ano="VB">identify</w>
    <w ano="DT">the</w>
    <w ano="NNS">cations</w>
    <w ano="IN">in</w>
    <w ano="JJ">known</w>
    <w ano="CC">and</w>
    <w ano="JJ">unknown</w>
    <w ano="NNS">samples</w>
    <w ano=".">.</w>
</s>
<s>
    <w ano="IN">By</w>
    <w ano="VBG">observing</w>
    <w ano="DT">the</w>
    <w ano="NNS">reactions</w>
```

Figure 3*: Excerpt from a morphologically annotated CHEMLAB corpus.*

3.1.3. Corpus summary and statistical data

Apart from the general corpus size, measured in the total number of running words, the compiled CHEMLAB and IASSES corpora are assessed by the following aspects:

- Corpus word distribution by morphological categories,
- Distinct possible action verbs and action category objects (single and multiword elements), that can be extracted from the corpus,
- Distinct possible multi-word keywords, extracted from the corpus.

Algorithms for the analysis of accumulated corpus data are presented in detail in project-related publication (Markievicz et al., 2013). The glossary of the most common action-verbs is obtained by building a verb-frequency list and filtering out the most frequent actions. In order to have a complete action representation, term-specific linguistic patterns, which include verbs, prepositional verbs (verb + preposition) and phrasal verbs (verb + [direct object] + adverb) and other multiword verbs (verb + direct object, verb + modifier) are used. Text preprocessing, leading to building of a glossary of possible action environment elements, involves collocation extraction methods. Collocation is a sequence of words that co-occur more often than it would be by chance (e.g. room temperature). Statistical log Dice coefficient method is used for extracting collocations from text. With action environment element glossary in place, classification of environment elements is done according to their action-specific roles by applying certain rules or search patterns.

The following software modules were prepared specifically for the ACAT project for analyzing the number of possible candidate-elements for process memory formation:

- for calculating word distribution by morphological categories (tags),
- for building frequency lists of single and multi-word terms.

Figure 4 and Figure 5 present the analysis results for word distribution by morphological categories in the general chemistry part of the CHEMLAB corpus. Figure 4 presents the accumulated percentage data distribution for the main morphological categories participating in the process memory formation – verbs, nouns, adjectives and prepositions. Figure 5 presents detailed data for word distribution by all available morphological categories.

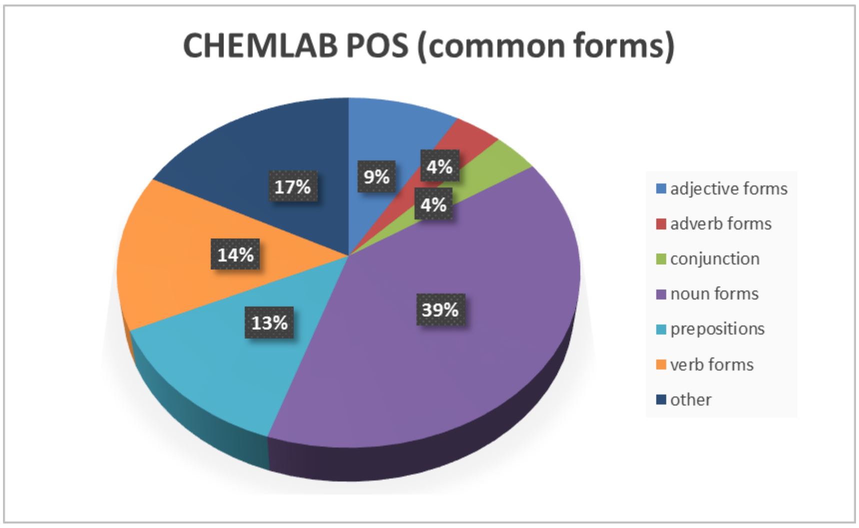


Figure 4*: The accumulated percentage data distribution for the main morphological categories of the general chemistry part of the CHEMLAB corpus.*

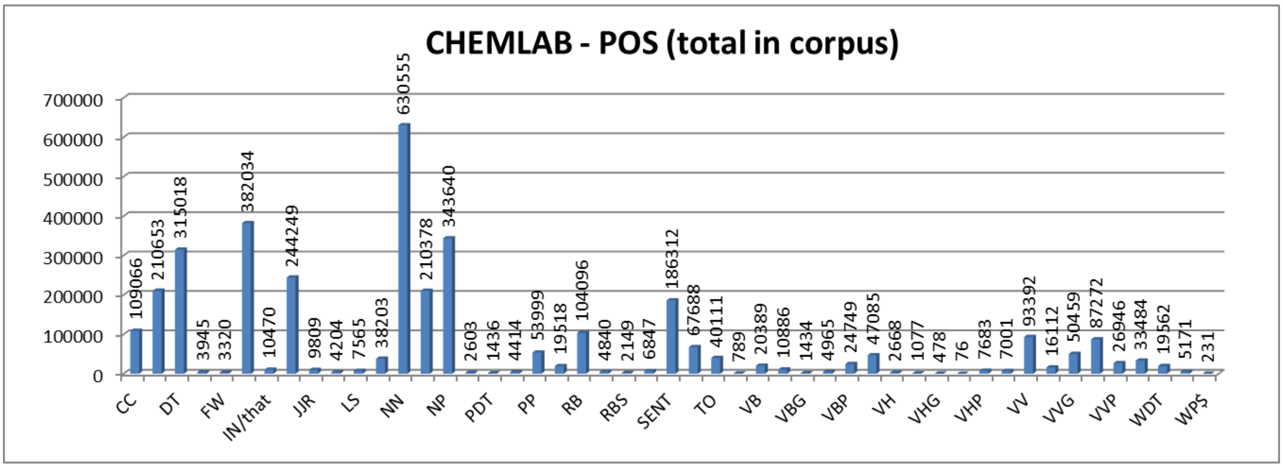
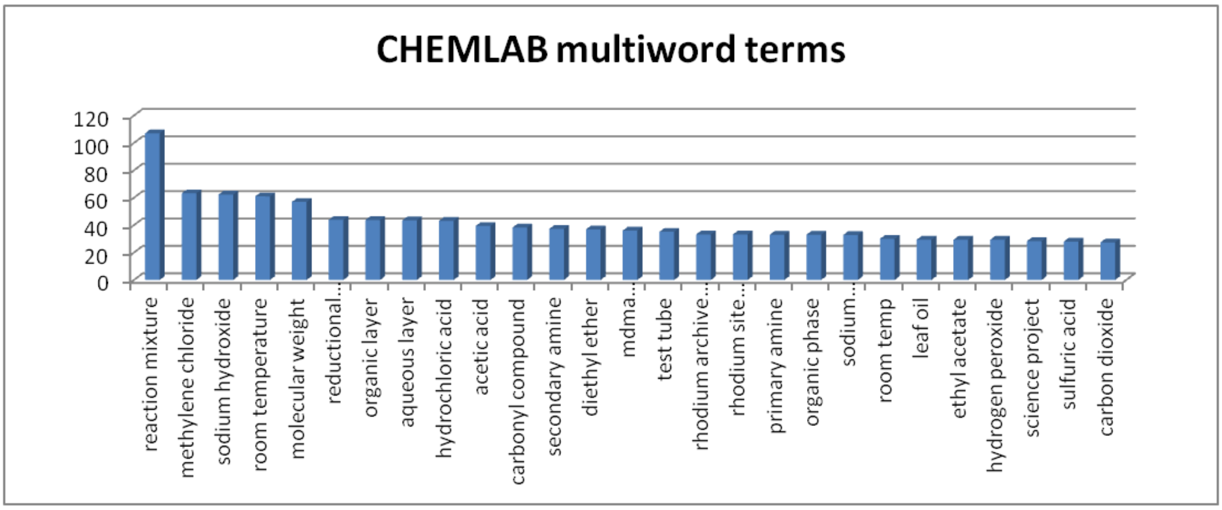


Figure 5*: Detailed data for word distribution by all the morphological categories of the general chemistry part of the CHEMLAB corpus.*

Figure 6 presents an excerpt from the results of extracting distinct possible candidates for process memory formation (multiword and single-word terms) for the general chemistry part of CHEMLAB corpus.



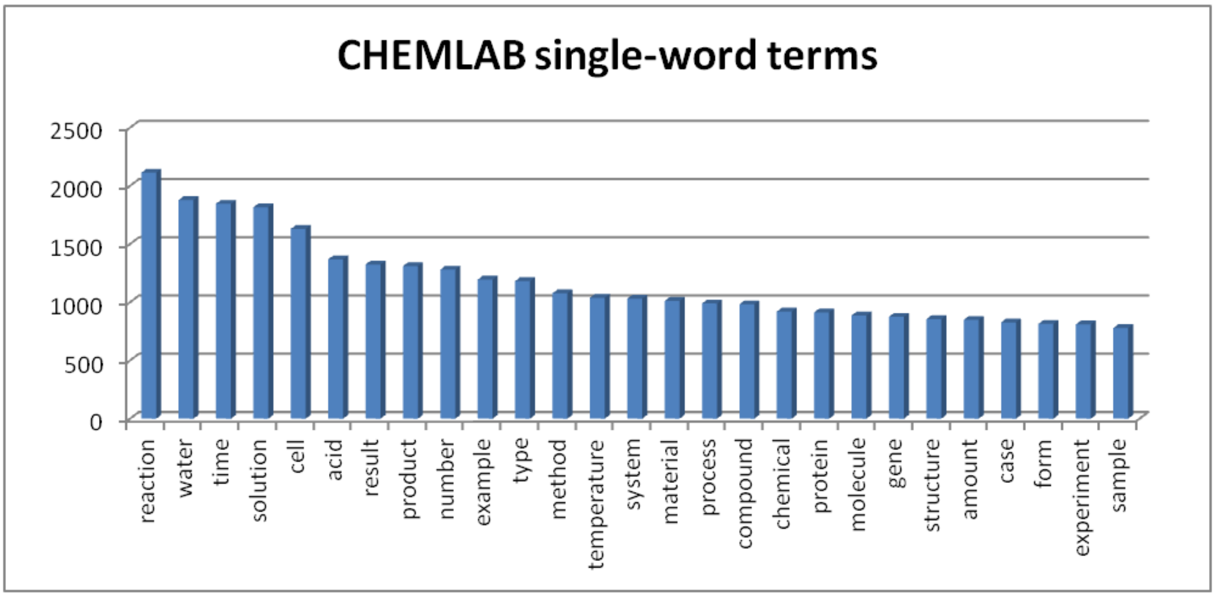


Figure 6*: Distribution of distinct possible single-word and multi-word candidates for process memory formation for the general chemistry part of the CHEMLAB corpus.*

Figure 7 and Figure 8, correspondingly, present the generalized and detailed analysis results for word distribution by morphological categories in the biochemistry and biotechnology part of the CHEMLAB corpus.

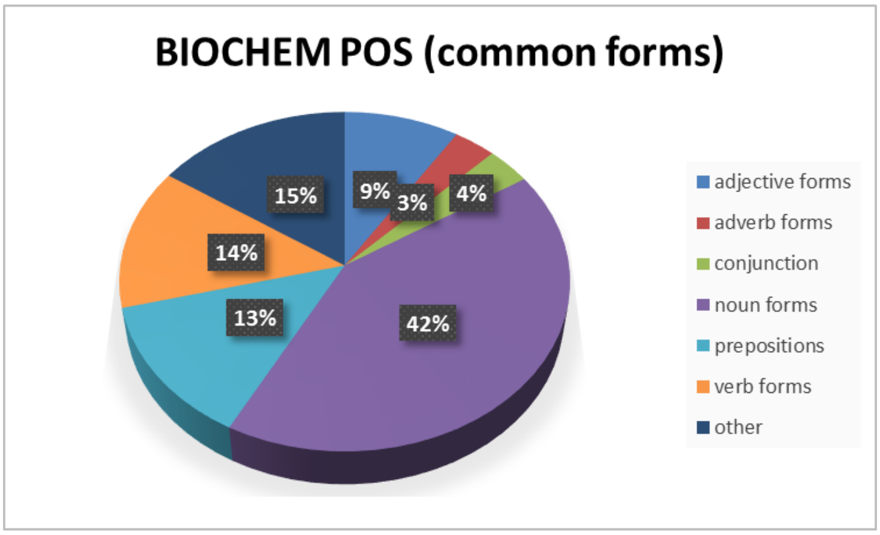


Figure 7*: The accumulated percentage data distribution for the main morphological categories of the biochemistry and biotechnology part of the CHEMLAB corpus.*

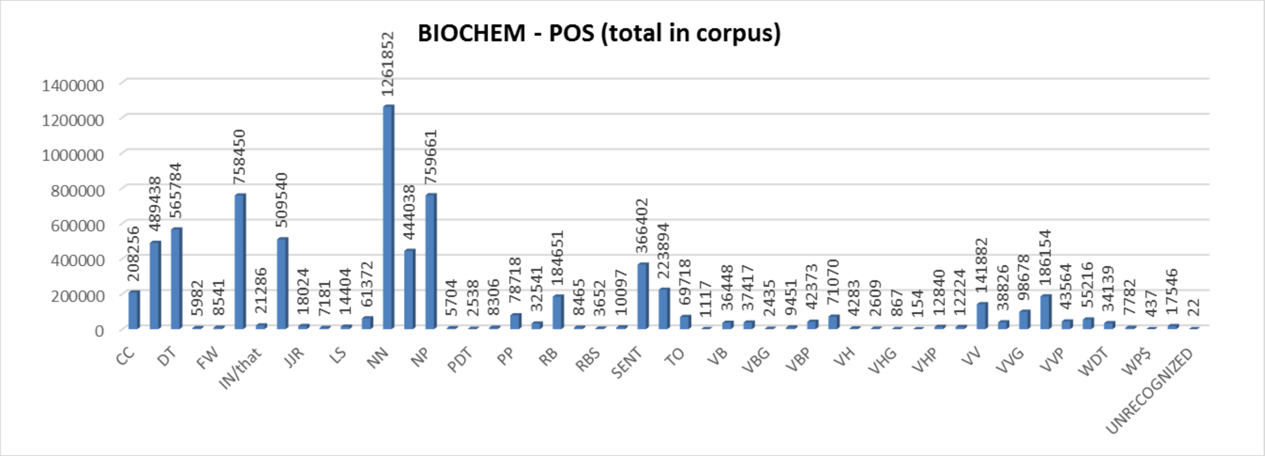


Figure 8*: Detailed data for word distribution by all the morphological categories of the biochemistry and biotechnology part of the CHEMLAB corpus.*

Figure 9 presents an excerpt from the results of extracting distinct possible candidates for process memory formation (multiword and single-word terms) for the biochemistry and biotechnology part of CHEMLAB corpus.

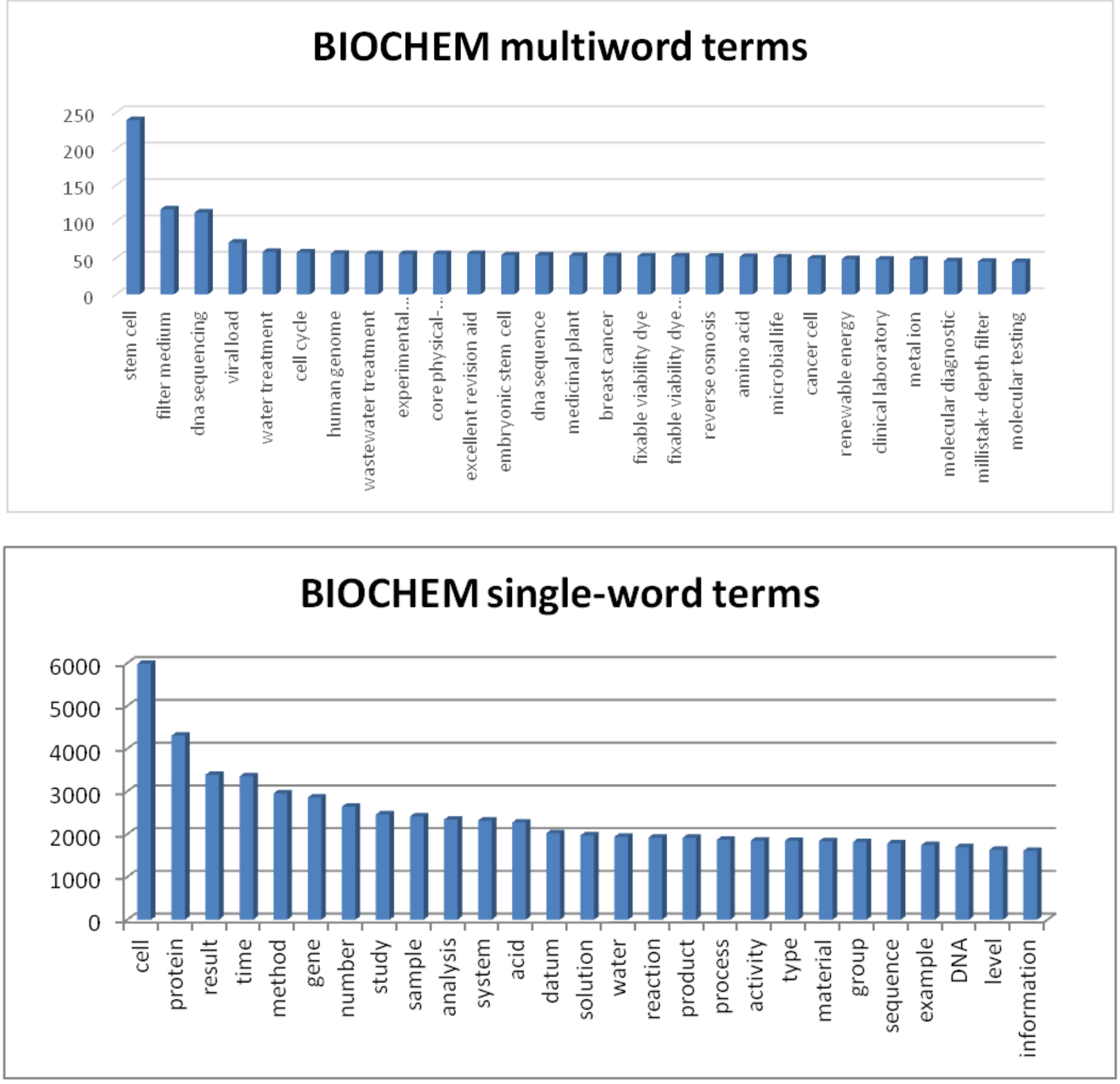


Figure 9*: Distribution of distinct possible single-word and multi-word candidates for process memory formation for the biochemistry and biotechnology part of the CHEMLAB corpus.*

Figure 10 and Figure 11 present the analysis results for word distribution by morphological categories in the in the IASSES corpus. Figure 10 presents the accumulated percentage data distribution for the main morphological categories participating in the process memory formation – verbs, nouns, adjectives, propositions. Figure 11 presents detailed data for word distribution by all available morphological categories.

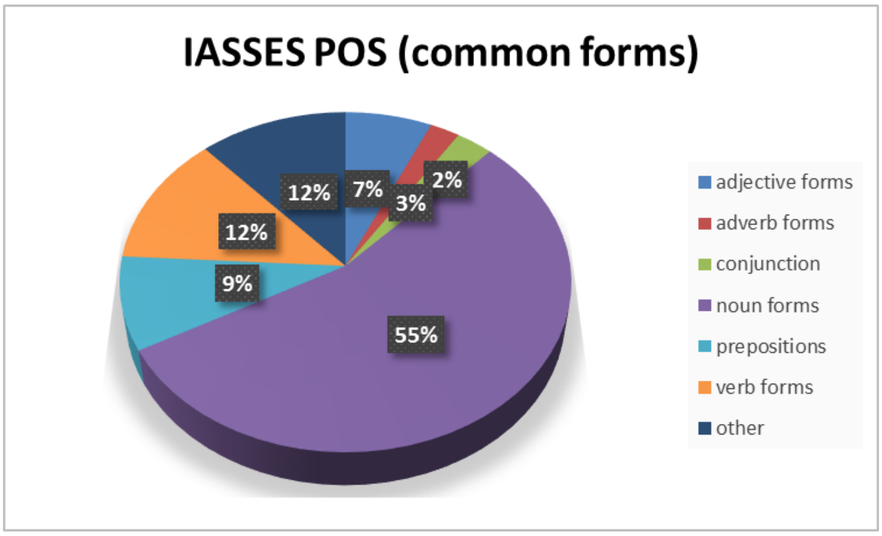


Figure 10*: The accumulated percentage data distribution for the main morphological categories of the IASSES corpus.*

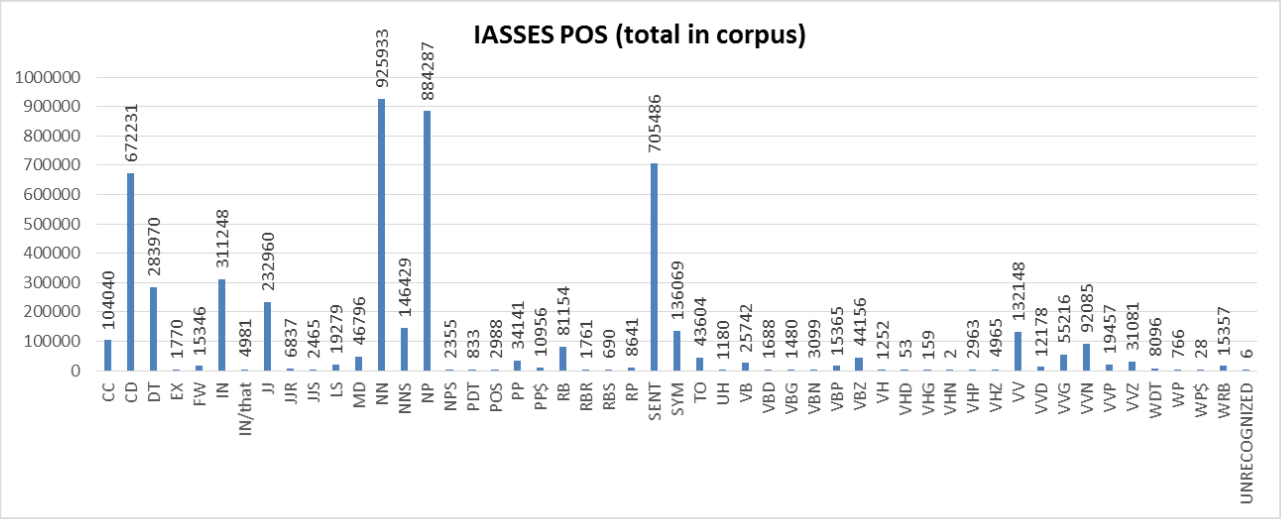
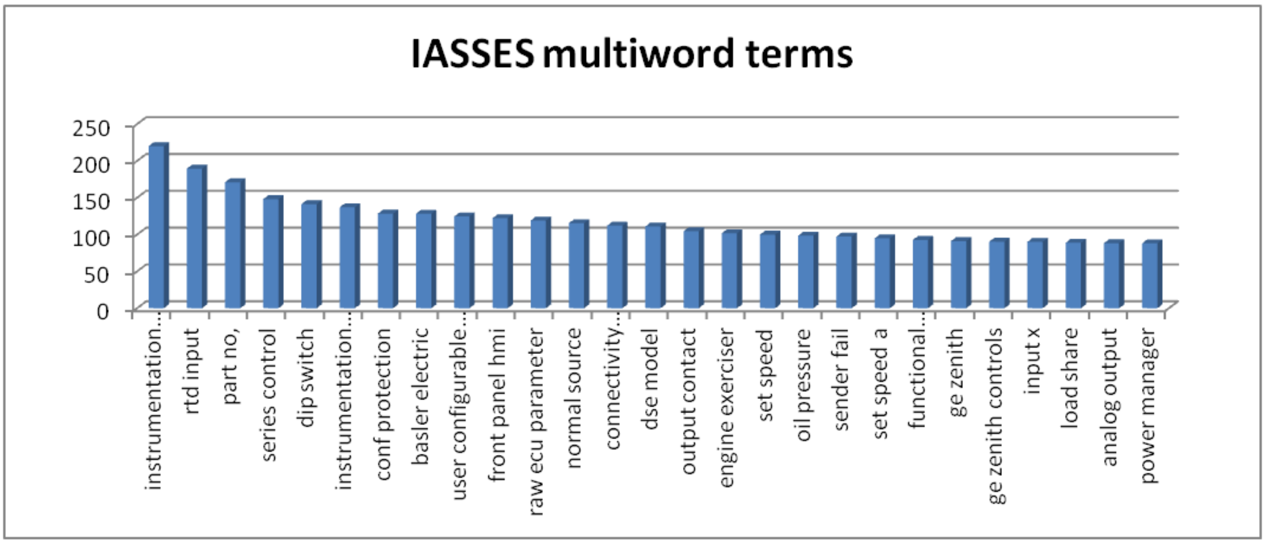


Figure 11*: Detailed data for word distribution by all the morphological categories of the IASSES corpus.*

Figure 12 presents an excerpt from the results of extracting distinct possible candidates for process memory formation (multiword and single-word terms) for the IASSES corpus.



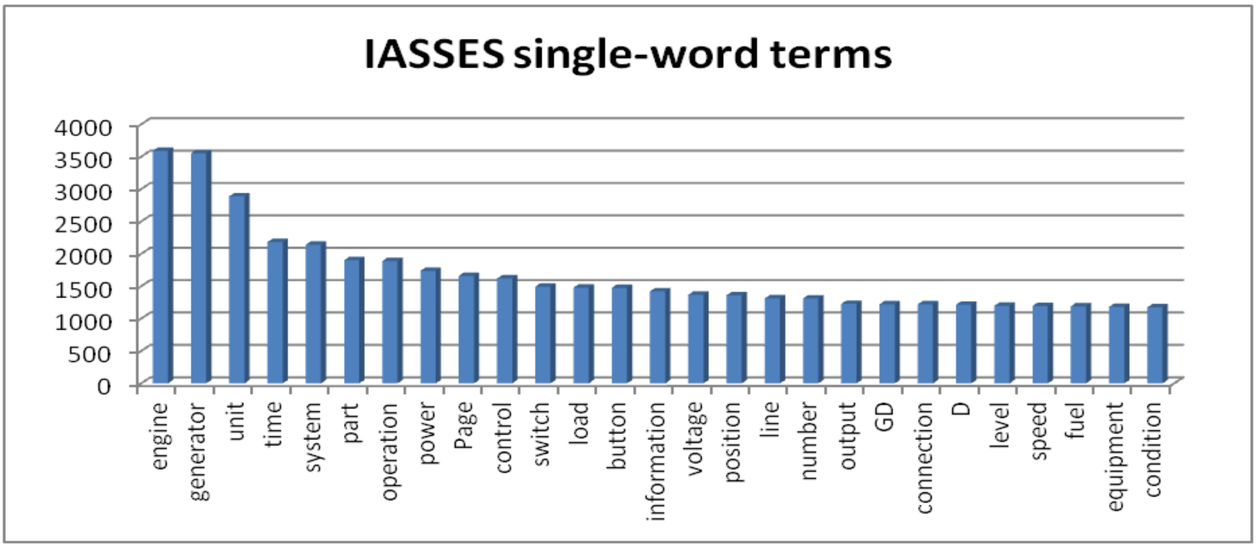


Figure 12*: Distribution of distinct possible single-word and multi-word candidates for process memory formation for the IASSES corpus.*

3.2. Formation of image databases

The main points, characterizing the formed image databases for CHEMLAB and IASSES scenarios of ACAT, are:

1) Procedures for image data acquisition and cleaning, data sources.
2) Image database structure, including metadata structure.
3) Summary for the accumulated image databases.

3.2.1. Procedures and data sources

Procedures for image data acquisition and cleaning and their sequence are presented in Figure 14.

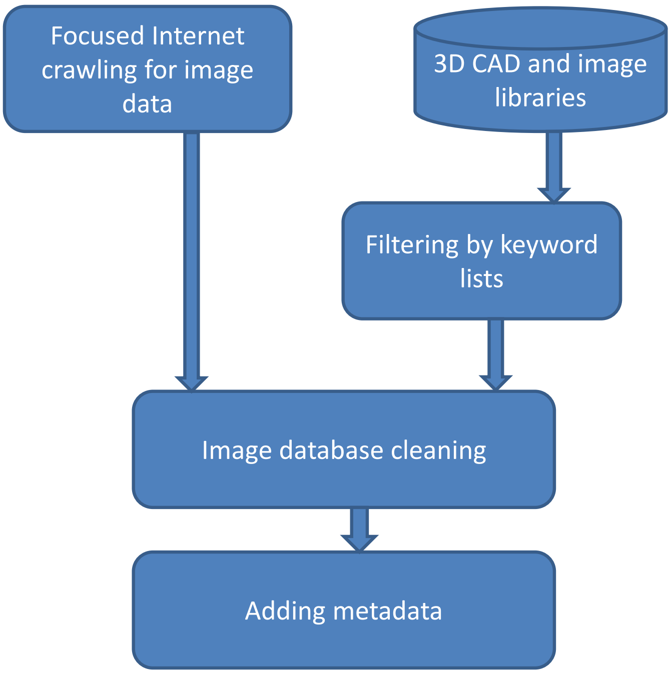

Focused Internet crawling for image data
3D CAD and image libraries
Filtering by keyword lists
Image database cleaning
Adding metadata

Figure 14: *Procedures for image database acquisition and cleaning.*

Both crawling of freely available Internet resources and use of 3D CAD and image libraries are employed for image data acquisition. Crawling is executed by applying image crawler, specifically built for ACAT, using domain-specific keyword lists. 3D CAD and image libraries employed include the GrabCAD Free 3D CAD library (http://grabcad.com/) and image libraries of project partners (SDU and UoB).

In the first processing layer, 3D CAD and image library data is filtered, using domain-specific keyword lists, accumulated by applying pre-analysis of domain specific texts and expert input.

Further on, the images are supplied to the second processing layer, dedicated to image cleaning by applying specialized image cleaning algorithms. Two algorithms (Kulvicius et al., 2014; Schoeler et al., 2014) for automated and unsupervised generation of “clean” image databases were developed which can cope with the problem of homographs, i.e., the words that are spelled the same but have different meanings. For example, the word “nut” could mean a hardware or a fruit. For disambiguation we make use of image searches (like Google), text searches and language translations. In the first approach (called SIMSEA, Kulvicius et al., 2014) we use additional linguistic cues to demarcate our intended meaning of a word. Here, we combine this linguistic refinement with the image-search in the following way. We conduct several different image subsearches, where we pair the basic search term with an additional linguistic cue. For example, if we are interested in the category “nut”, we search for “bolt nut”, “metal nut”, “plastic nut”, etc., depending on the context we are interested in. The expectation is that images that are retrieved by more than one of these subsearches are more likely to be of interest, than those that are retrieved only once. In the second approach (called TRANSCLEAN, Schoeler et al., 2014), in order to address the problem of homographs, we present a method for automatic (without human supervision) generation of task-relevant training sets for object recognition by using the information contained in a language-based command like “tighten the nut”. We ground our approach based on two facts: 1) homographs rarely occur for one word in multiple languages at the same time and 2) context information (action) provided by the command can be used in order to get rid of ambiguous and non-task relevant translations. We evaluated performance of our methods on image classification task (10/15 ambiguous classes) and obtained on average 17% and 23% (SIMSEA and TRANSCLEAN, respectively) improvement in object recognition as compared to standard Google search. The details of presented cleaning algorithms can be found in project-related publications (see Attached papers).

Finally, metadata, describing the classification, format and image quality information for the acquired images, is attached. This is done by adding a satellite XML file for each image object, using a semi-manual procedure and proprietary software.

3.2.2. Image database structure

The ACAT image databases are structured in the following way:

- CHEMLAB image database, containing different images of tools for chemical experiments;
- IASSES image database, containing objects for robot manipulation in the IASSES scenario as well as assembly instruction illustrations.

Sources for the CHEMLAB image database are:

- Partner supplied images (UoB);
- Images from free CAD libraries (GrabCAD);
- Images crawled from Internet.

Depending on the format and quality of the acquired images, the IASSES image database is structured in the following way:

- Images crawled from Internet,
- Partner provided rendered images based on CAD models (SDU),
- Partner provided images from rotor caps (SDU).

The partner provided IASSES scenario images illustrate four different sequences - failing and succeeding grasping actions with two different orientations: gripper top and gripper front (Table 2). For each distinct image, the following versions are given: low-resolution image from the carmine sensor, high-resolution image from the stereo pike cameras and high-resolution images from the stereo pike cameras with addition pattern projection.

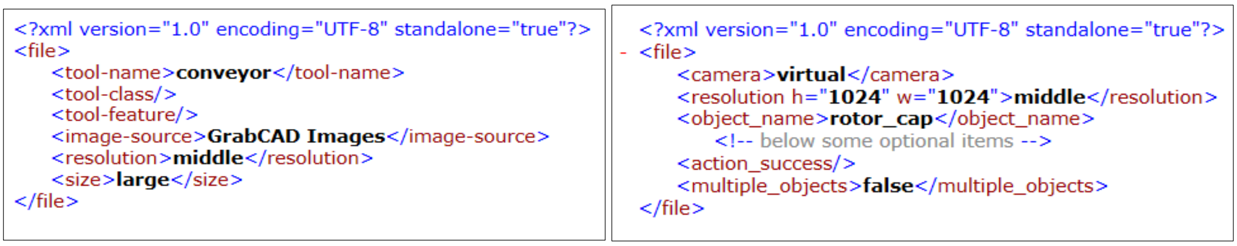
Table 2*: IASSES scenario images, illustrating the success/failure of grasping action*

GRIPPER ORIENTATION	ACTION FINAL STATE	IMAGE SOURCE	SIZE (MB)	FILES	RESOLUTION
gripper top	failure	stereo projector (png)	927,4	208	high
		stereo (png)	466,5	116	high
		carmine top	21,1	24	low
	success	stereo projector (png)	839,4	190	high
		stereo (png)	501,8	124	high
		carmine top	14,9	17	low
gripper front	failure	stereo projector (png)	765,2	180	high
		stereo (png)	480,3	120	high
		carmine top	20,2	23	low
	success	stereo projector (png)	429,1	100	high
		stereo (png)	110	27	high
		carmine top	20,2	23	low

The rendered images based on CAD models are provided for illustration purposes only - as the models do not contain information about material or texture, the generated images are not photo-realistic. Currently the CAD models are utilized to extract 3D shape information which in turn is used to identify the object and its pose in real-world situations. The models consist of: magnet, rotor core, rotor cap, rotor axle, pressure ring.

For both CHEMLAB and IASSES image databases, each image file is accompanied by a satellite XML file with object-specific metadata, describing the classification, origin and image quality information.

Figure 15 presents examples of XML files, showing the structure of image metadata.



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<file>
    <tool-name>conveyor</tool-name>
    <tool-class/>
    <tool-feature/>
    <image-source>GrabCAD Images</image-source>
    <resolution>middle</resolution>
    <size>large</size>
</file>
```

```
  <?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <file>
      <camera>virtual</camera>
      <resolution h="1024" w="1024">middle</resolution>
      <object_name>rotor_cap</object_name>
          <!-- below some optional items -->
      <action_success/>
      <multiple_objects>false</multiple_objects>
  </file>
```

Figure 15*: Examples of XML files with image metadata for CHEMLAB and IASSES image databases*

3.2.3. Image database summary

The compiled CHEMLAB and IASSES image databases are assessed by the following size/quality aspects:

- total number of images,
- image distribution by sources of acquisition,
- image distribution by image file formats.

The size of the accumulated CHEMLAB image database is 1818 images. Distribution by image sources for CHEMLAB is:

- 1568 images (86%) – crawled from Internet (Google images),
- 234 images (13%) – obtained from free CAD libraries (GrabCAD),
- 16 images (1%) – high quality partner-supplied images (UoB).

Distribution by image formats for CHEMLAB is:

- 1566 images (86%) – JPEG images,
- 237 images (13%) – bitmap images (BMP, PNG, GIF),
- 15 images (1%) – other formats (SVG, TIFF, SLDPRT, etc.).

The accumulated IASSES image database contains 1208 images. Distribution by image sources for IASSES is:

- 50 images (4,14%) – crawled from Internet (Google images),
- 1152 images (95,36%) – partner provided images from rotor caps (SDU),
- 6 images (0,5%) - partner provided rendered images based on CAD models.

Distribution by image formats for IASSES image database is:

- 43 images (3,74%) – JPEG images,
- 87 images (7,24%) – PPM files,
- 1067 images (88,77%) – PNG images,
- 3 images (0,25%) – GIF images.

Figure 16 and Figure 17 present the examples of obtained different CHEMLAB and IASSES images correspondingly.

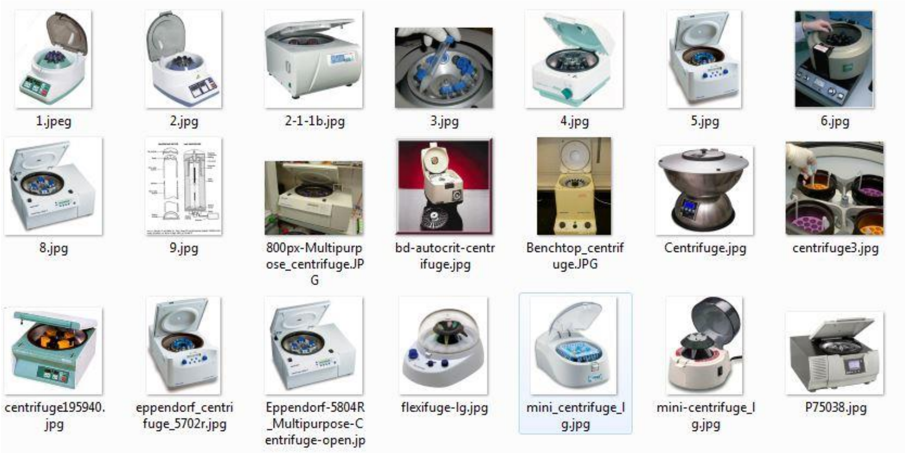


Figure 16*: Examples of obtained CHEMLAB images for tool 'centrifuge'*

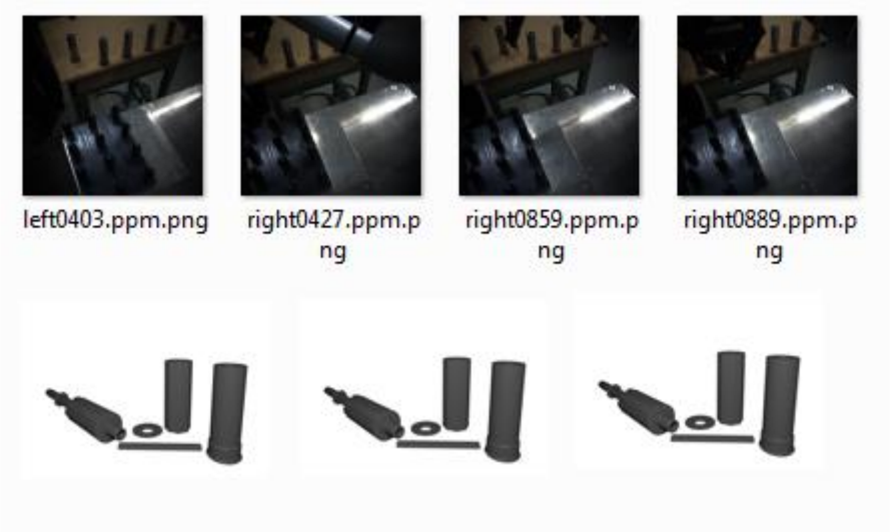


Figure 17*: Examples of accumulated IASSES images*

4. Conclusions

The corpora compiled for the CHEMLAB and IASSES scenarios, as well as the image databases, will serve as the basis for process memory formation through the identification of action categories and action background elements.

The information, collected for domain-specific ACAT corpora and databases, together with data structures defined in deliverable D2.1 on ”Background data structure”, make it possible to reach the ACAT project Milestone MS2 ”Information infrastructure prepared & data structures defined“.

Both the corpora and databases will be further continuously updated along with the Project timeline, and will be presented in an updated form in the deliverable D1.4.

5. Attached papers

Kulvicius, Tomas , Schoeler, Markus, Markelic, Irene, Tamosiunaite, Minija and Wörgötter. Florentin. Semantic Image Search: Automated Generation of Image-databases for Robotic Applications, 2014. Submitted to the journal „Robotica“.

Markievicz, Irena, Vitkute-Adzgauskiene, Daiva and Tamosiunaite, Minija. Semi-supervised Learning of Action Ontology from Domain-Specific Corpora. Information and Software Technologies, Springer, Berlin Heidelberg, 2013. 173-185.

Schoeler, Markus, Wörgötter, Florentin, Aein, Mohamad Javad and Kulvicius, Tomas. TRANSCLEAN: Unsupervised generation of training-sets for visual object recognition in robotics based on multi-language cues, 2014. Submitted to the IROS 2014 conference.

Semantic Image Search: Automated Generation of Image-databases for Robotic Applications

Tomas Kulvicius* Markus Schoeler Irene Markelic Minija Tamosiunaite
Florentin Wörgötter

January 17, 2014

Abstract

Learning and generalization in robotics is one of the most important problems. New approaches use internet databases in order to solve tasks and adapt to new situations. Modern search engines can return a large amount of information according to a query within milliseconds. However, not all of the returned information is task relevant, partly due to the problem of homonyms and polysemes. Here we specifically address the problem of automated generation of context-dependent image-databases for robotic applications by using internet image search. We suggest a bi-modal solution, combining visual and textual information, based on the observation that humans use additional linguistic cues to demarcate intended word meaning. We evaluate the quality of our approach by comparing it to human labelled data and also in object classification experiment. We find that, on average, our approach leads to improved results in comparison to plain Google searches, and that it can treat the problem of homonyms and polysemes.

Keywords: Internet based knowledge, Homonyms/Polysemes, Semantic search, Image database cleaning

1 Introduction

Humans can learn and generalize to new tasks very quickly whereas for robots this is still not an easy task which makes it one of the most important and relevant problems in robotics. One of the most common approaches in learning and generalization is learning from previous experiences [32, 25, 19, 20]. Some new approaches use internet databases in order to adapt/generalize to new situations [31, 2, 30]. For example, robot can search internet databases for images of objects in order to recognize the objects appearing in the scene. In particular, here we are interested in generation of "clean" (context-dependent) image databases for robotic applications by using internet image search. Although modern search engines like Google or

*Georg-August-Universität Göttingen, Bernstein Center for Computational Neuroscience, Friedrich-Hund Platz 1, DE-37077 Göttingen, Germany, E-mail: tomas@physik3.gwdg.de

Yahoo do an amazing job in returning a large number of images according to a query within milliseconds, not all of the returned images are task/context-relevant. A reason for spurious results is that most image searches rely on text-based queries, which is justified, since visual and textual information are dual to some degree. An *image* of a cup can be interpreted as the visual representation of the concept cup, whereas the *word* cup can be seen as a linguistic handle to the concept cup as represented in the human mind [13]. Therefore, existing tools for text-based information retrieval applied to image search can lead to relatively good results [7]. Problems arise mainly due to ambiguities: 1) The same linguistic handle can map to several, different concepts, e.g., homonyms and polysemes. Homonyms are words that are spelled and pronounced the same, but have different meanings. Polysemes refer to same spelling words with different but related meanings. An example is the just mentioned word "cup" which can refer to a cup as used for drinking (e.g., in robotic breakfast scenario), as well as the to the cup as a trophy, e.g., in a Soccer World Cup. Without any further information, e.g., contextual information, it is not possible to infer which domain is actually referred to. 2) Text-based image search relies on the assumption that textual information that is somehow related to an image, e.g., text placed close-by an image on a web page refers to the image content [7]. This assumption is reasonable, however not always correct, e.g., not every web-page designer/programmer names images according to their content.

A lot of effort has been spent on trying to resolve the problem of obtaining unclean image search results, often with the goal of object detection or image categorization, by making additional use of image content in form of visual cues, e.g., features like local image patches, edges, texture, color, deformable shapes, etc. [11, 12, 10, 14, 3, 28, 18, 16, 34, 17, 1]. All these approaches use textual information, too. Either implicitly by using the results of text-based image search engines e.g., [10, 12], or constructing their own image search [28, 3, 1], or explicitly, by making use of image tags and labels as found in photo-sharing websites like Flickr [14, 34, 3]. An interesting work is [34], because it is inverse to the standard procedure. Instead of using images with similar text labels to obtain image features for classification, they reverse the problem and use similar images to obtain textual features.

To our knowledge all of the aforementioned approaches achieve an improved precision of the result set, however, none can automatically cope with the problem of homonyms and polysemes which would be required in automated robotic applications like [2, 30]. For example in [12] a re-ranking of images obtained from Google searches was proposed, based on the observation that images related to the search are visually similar while unrelated images differed. This "visual consistency", what we will here call inter-image similarity, was measured using a probabilistic, generative image model, and the EM-algorithm was used for estimating the model parameters from image features. Naturally, due to the underlying assumption, this will not work well for homonyms, since for these many images that are actually closely related to the search can have a very different appearances. A similar problem was faced in [10], where an extended version of pLSA (probabilistic Latent Semantic Analysis) was used to learn a clustering of images obtained from a Google search. A solution suggested in [3] copes with the polysemes problem but requires human supervision for this stage. Google text search is used to collect webpages for 10 animals. Then LDA (Latent Dirichlet Allocation) is applied to text from these pages to discover a set of latent topics. Images extracted from the webpages are then assigned to the identified topics, according to their nearby word likelihood. The problem of polysemes is tackled by a human user who manually selects or rejects these image sets.

Here, we present a novel approach which we call SIMSEA (Semantic IMage SEArch) which also aims at increasing the precision of Internet image search results. Its most prominent advantage is that it can cope near-to automatically with polysemes and homonyms. This is achieved by exploiting the fact that also humans need to resolve ambiguities in every-day speech, e.g., we may say "the bank - that you can *sit* on" to distinguish it from the bank that deals with money. Thus, we give additional cues to demarcate our intended meaning of a word. Here, we combine this linguistic refinement with the image-level in the following way. We conduct several different image searches, where we pair the basic search term with an additional linguistic cue. For example, if we are interested in the category "cup", (the basic search term, e.g., in some robotic breakfast scenario), we search for "coffee cup", "tea cup", etc. The expectation is that images that are retrieved by more than one of these subsearches are more likely to be of interest, than those that are retrieved only once. Note that for simplicity, in this paper we defined additional cues manually. In general, automated extraction of object descriptors (cues) can be done using methods of natural language processing [8, 26, 22], however, this is out of the scope of the current paper.

We evaluate the quality of SIMSEA algorithm by comparing image sets returned by SIMSEA to human labelled data. Additionally we test SIMSEA's performance on image classification where we used images obtained by the SIMSEA algorithm as training set for a classifier and compared to the classification results where training data was retrieved by plain Google search. In our evaluation, images sets are everyday kitchen objects, as we are having in mind robotic kitchen scenarios that are frequently used as test cases in current service robotic research.

The paper is structured as follows. First, we give a detailed description of SISMEA procedure in section 2, followed by the explanation of how we evaluated our method and the presentation of the achieved results in section 3. Finally we discuss and conclude our work in section 4.

2 SIMSEA Algorithm

The approach is summarized in Fig. 1 A and the details of its stages, which are enumerated in the figure, are described below. The goal is to find "clean" results for image searches with respect to given task/context, which later can be user for object learning, recognition and generalization.

To achieve the above stated goal, given a basic search term (see Fig. 1, step 1), e.g., "glass", we

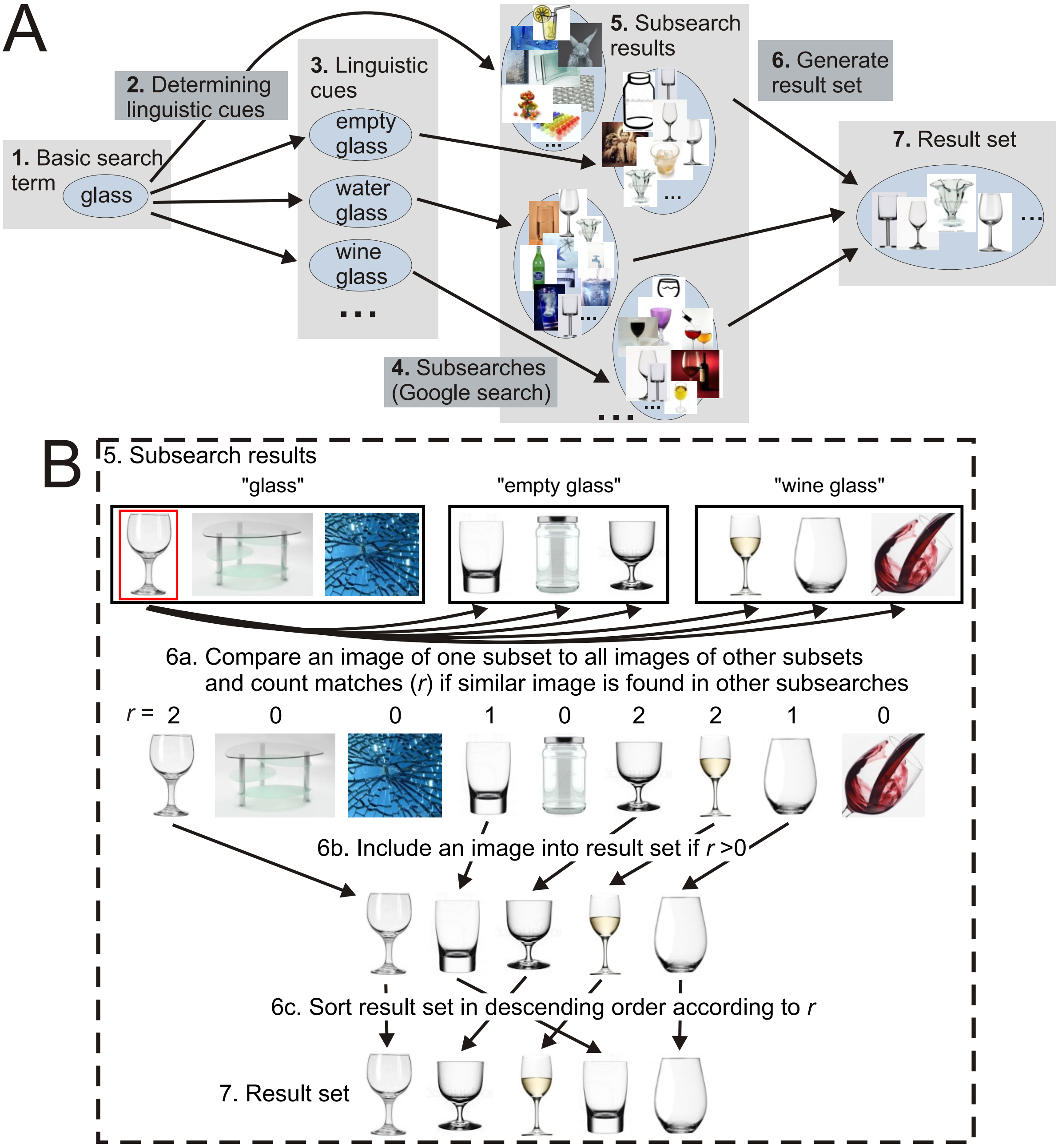

A
1. Basic search term
glass
2. Determining linguistic cues
3. Linguistic cues
empty glass
water glass
wine glass
...
4. Subsearches (Google search)
5. Subsearch results
6. Generate result set
7. Result set
B
5. Subsearch results
"glass"
"empty glass"
"wine glass"
6a. Compare an image of one subset to all images of other subsets and count matches (r) if similar image is found in other subsearches
r = 2 0 0 1 0 2 2 1 0
6b. Include an image into result set if r >0
6c. Sort result set in descending order according to r
7. Result set

Figure 1: **A)** Procedure of SIMSEA algorithm exemplified on the category “glass”. **B)** Generation of result set (step 6).

determine several linguistic cues (step 2), e.g., “empty”, “water”, “wine”, etc. In general, linguistic cues can be any, as long as they are from the specific context we are interested in. For the sake of simplicity, in this paper we defined linguistic cues manually, but it can also be done using methods of natural language processing or any other method. [8, 26, 22]. As a result, we obtain the list of linguistic cues + basic search term (step 3), which will be used to perform Google search.

After generation of linguistic cues, we conduct several image searches to which we refer as *subsearches* (step 4), see Fig. 1 A. A subsearch is conducted using the *basic search term* (step 1) with additional *linguistic cues* (step 3). E.g., if interested in the category “glass”, we search for “glass”, “emty glass”, "water glass", "wine glass", etc., using Google search. The set of images retrieved by a subsearch is consequently referred to as *subsearch result* (step 5). The expectation is that images that are retrieved by more than one subsearch are more likely to be task/context-relevant than those that do not. These images form the final *result set* (step 7). Note that we do not consider only images that have exact copies in other subsearch result sets, but instead relax this demand and also consider images as relevant if merely a similar image is returned by another subsearch.

The generation of result set (step 6) is graphically represented in Fig. 1 B and procedure is as follows. We take an image I_i^k from a subsearch $k\,(k = 1 \dots m)$ and compare it to all other images I_j^l of other subsearches $l\,(l = 1 \dots m)$ and count matches r_i^k if similar images are found in other subsearches (step 6a). Note that we do not compare to the images of the subsearch itself. This is because we are not interested in intra-subsearch similarity due to the following reason. We may receive many images of the same topic during one search but which are unrelated to what we are interested in. The pseudo-code of the result set generation procedure is given in Fig. 2.

```
Get images $I_i^k$ $(k = 1 \dots m, i = 1 \dots n_k)$, where
$m$ is the number of subsearches and
$n_k$ is the number of images in a subsearch $k$;
Set similarity threshold $\theta$;
Initialize matches $r_i^k = 0$.
FOR $k = 1$ to $m$
    FOR $i = 1$ to $n_k$
        FOR $l = 1$ to $m$
            IF $k! = l$
                FOR $j = 1$ to $n_l$
                    Compare images $I_i^k$ and $I_j^l$ by
                      computing distance $d_i^k$
                      in some metric space;
                    IF $d_i^k < \theta$
                        $r_i^k = r_i^k + 1$;
                        break.
```

Figure 2: A pseudo-code for the generation of the result set (steps 6a and 6b; see Fig. 1 B).

In general, in order to compare images one can use any kind of features and any kind of metric (distance measure). In this paper, in one case we used "Bag-of-Words" approach and Hellinger distance, whereas in the other case we computed correlation coefficient between gray-scale images. For details please see section 3.

We include an image I_i^k into result set (step 6b) if $r_i^k > 0$, i.e., if a similar image appeared in at least one of other subsearches, too. And finally, we rank and sort the retrieved result set (step 6c) according to matches r in a descending order and, this way, obtain the final result set (step 7). The ranking is supposed to indicate how relevant a given image is, e.g., a glass image with a high ranking factor should be considered to be very likely a true representative of the category glass, whereas an image with a low ranking factor can be considered to be very likely not a good representative of its class. Note that we delete duplicated images from the final result set.

3 Evaluation

3.1 Comparison to Human Data

In the first phase, as a proof of concept, we validated SIMSEA performance, by comparing images returned by SIMSEA algorithm, to the human labelled data. We expected that images obtained by SIMSEA method will more closely match human data compared to those returned by plain Google search due to problem of polysemes/homonyms as discussed above.

3.1.1 Methods

We investigated four different categories (basic search terms) taken from a kitchen scenario: cup, glass, milk and apple. Cup is a polyseme: drinking-cup or football-cup; glass and apple are homonyms: vision-aid, drinking-glass, and glass as a material; or brand-apple and fruit. Milk is a special case, because as a liquid it usually comes in some kind of container, e.g., tetra-pak, glass, bottle, cup, etc.

For each of the four categories we conducted a varying number of subsearches in which we combined the basic search term with an additional linguistic cue as described above. The linguistic cues for subsearches are given in Table 1.

Table 1: Linguistic cues for Google subsearches used for comparison to human data.

Basic search term	Linguistic cues
cup	coffee, tea, full
apple	delicious, green, red, ripe, unripe, sour, sweet
milk	cold, hot, fresh, healthy, tasty
glass	empty, full, juice, milk, water, wine

To be able to measure inter-image similarity we used a “Bag-of-Words” approach. In such an approach each image is represented by a histogram over a fixed number of so-called “visual words” which are also often referred to as “codebook”. First, the codebook needs to be generated. For that we used a small, randomly chosen subset of 40 images, from each category. We computed Pyramid Histogram of Visual Words (PHOW features,[5, 6]) for all these 160 (40 × 4 categories) images which we then quantized into K vectors - the visual words - using K-means clustering. In thus study we set $K = 200$. PHOW features are state-of-the-art image descriptors based on a variant of dense SIFT [21]. In this method, a grid with a defined spacing (here we used 5 pixels) is laid over an image and at each grid point four SIFT descriptors, varying in radii to allow for scale variations, are computed. This can be done on various levels of “Pyramid”, but here we suffice with the first level, thus, to be precise we were actually using HOW descriptors [5, 6]. We used the VLFeat library [33] to compute the HOW descriptors and the subsequent vector representation of the images.

To compute the similarity between image pairs we used the Hellinger distance. The Hellinger distance between two distributions P and Q is denoted $H(P,Q)$ and satisfies $0 \leq H(P,Q) \leq 1$ (where one denotes large distance and zero denotes identical images). It is defined as follows:

$$H(P,Q) = \sqrt{1 - BC(P,Q)}, \tag{1}$$

where BC denotes the Bhattacharyya coefficient which, in the discrete case, is defined as:

$$BC(P,Q) = \sum_{x \in X} \sqrt{P(x)Q(x)}. \tag{2}$$

Here X denotes the common domain over which the two distributions are defined. We define two images to be similar if their Hellinger distance is below a fixed threshold θ. In this study we used $\theta = 0.15$ (experimentally chosen).

Since the goal is to find a subset of images which meets the semantic expectation of the user, we need some "ground truth", i.e., a set of true samples, to evaluate our algorithm. For this issue we let five human subjects classify the same data that was input to the algorithm according to the given categories. This way we can gather various subjective human opinions and determine those images that get assigned the same labels by all subjects and also those where opinions differ. In the following we describe the ground truth retrieval procedure.

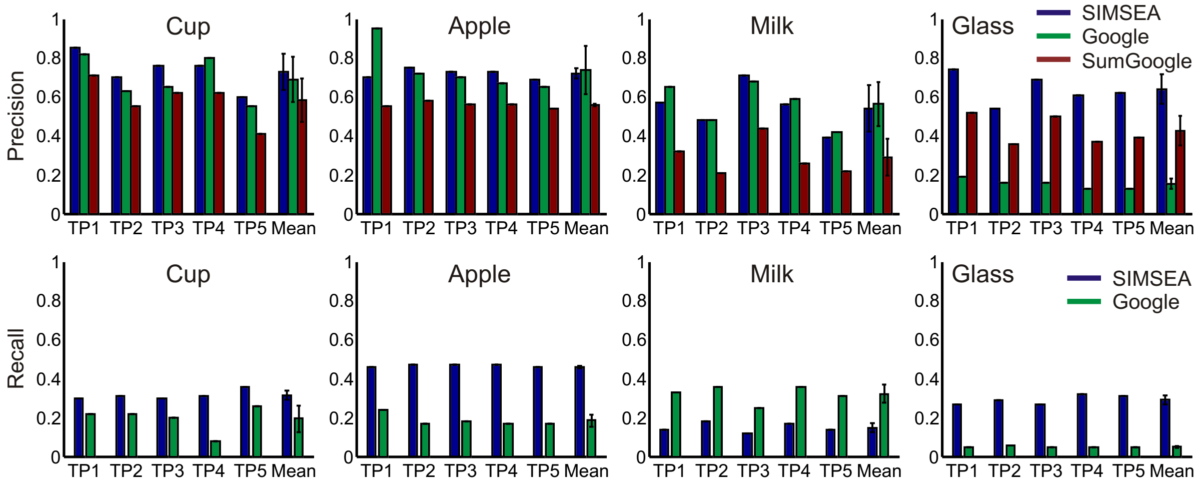


Figure 3: Precision and recall of SIMSEA, a standard Google search (Google) and the cumulative data from all subsearches for a given category (SumGoogle) with respect to the data obtained from each test person (TP1-5) for the categories. The vertical errorbar for the mean indicates the variance. Note that the recall for SumGoogle is always one and is not shown.

Each human was instructed to decide for each image from the subsearches for specific category whether it belonged, in his/her opinion, to the category or not. To make this evaluation as fair as possible, all humans were given precisely the same information by means of an instruction. The subjects were told that there are four categories and that they are from a kitchen scenario, thus, glass was supposed to be for drinking, and not for aiding vision, etc.

We assess the quality of the algorithm by computing precision and recall on its output, see Eq. 3, with respect to the ground truth data from each human subject:

$$\begin{aligned} \text{precision:=}&(A \cap B)/|A| \\ \text{recall:=}&(A \cap B)/|B|, \end{aligned} \tag{3}$$

where A is the set of retrieved samples and B is the set of true samples, i.e., in our case A is the set of samples retrieved by an algorithm and B is the set of samples belonging to a given category selected by each human subject. Since there were five human subjects, there are five true sample sets, with respect to which we compute precision and recall.

3.1.2 Results

The results of comparison to human labelled data are given in Fig. 3 where we compare three different seaches: 1) images obtained by SIMSEA algorithm, 2) images returned by a standard Google searches (Google) and to 3) the union of all subsearches of a given category (SumGoogle). For the case 2, we conduct standard Google searches with the basic search terms for each category, e.g., for the category glass, the set A (see Eq. 3) is the set of images returned by a Google search using the search term "glass". For the case 3, we set A to the union of the images from all subsearches of a given category. Note, that for SumGoogle the recall is always one. This is because the ground truth set from all human subjects is a subset of the union of subsearches for a category, in other words $B \subset A$.

To be useful, precision and recall of SIMSEA should be higher than those of the standard Google search and SumGoogle. In other words, most human subjects should find that the output of SIMSEA gives more relevant results than the Google standard search and SumGoogle (precision), and also that SIMSEA returns more of the overall available relevant samples (recall). It can be seen from Fig. 3 that except for the category milk SIMSEA indeed outperforms the standard Google search and SumGoogle.

For the category cup, in terms of precision, almost all humans except the test person 4 (TP4) agree more with the results of the SIMSEA algorithm. It can also be seen that the values for precision and recall differ between the subjects which shows, what we had already expected, that assigning images to a certain category also depends on subjective opinions. For the category apple, TP1 shows a very clear preference for the Google search results. Due to TP1 also the precision is higher for the Google search than the automatic routine. However, TP1's opinion is not in accordance with that of the other subjects, which all have a precision value around 0.7 and therefore we consider this to be an outlier. Without TP1's influence SIMSEA outperforms the Google search for "apple", too. For the category milk we can observe

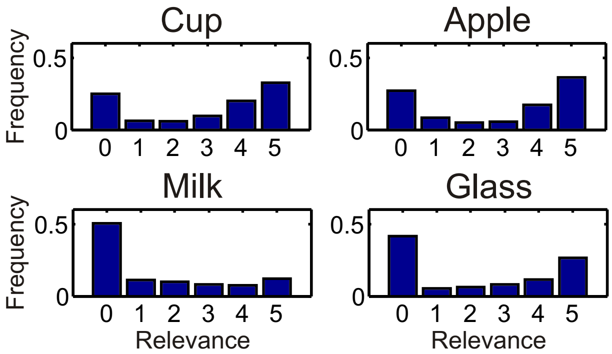


Figure 4: Histogram of image category membership assigned by the five human subjects from which we derive the image relevance.

a different case, most human subjects are more in accordance with the results of the Google standard search. A possible reason for that can be found in Fig. 4 where we show a histogram indicating for each category how many of the test persons considered each given image as being member of a category. Since there were five test persons each image can be selected as category member between zero and five times. We assume that images which were considered by none of the test persons as category member should be assigned the lowest relevance, and vice versa, images considered by all test persons should be assigned the highest relevance. We see that for all categories there are clear peaks for images that all human subjects consider as category member and for those that all human subjects consider to not be category members, except for the category milk. Here, there is no peak at 5, which means that there is no clear agreement among subjects what milk is in presented images. This might be due to the fact that, as we have already stated, milk as a liquid is depicted to be contained in different kind of containers. We can assume that for this reason SIMSEA is not performing well for this category either. For the category glass, there was a strong agreement among all subjects, and in this case SIMSEA ouperformed Google search.

3.2 Image Classification

In addition to comparison to human labelled data we also tested SIMSEA's performance in object classification experiment, where subjectiveness is excluded and we directly evaluate whether SIMSEA algorithm can improve object classification in robotic scenarios. Here we specifically selected ten different classes from kitchen scenario where words (basic search terms) have several different meanings (see Table 2).

3.2.1 Methods

As in previous experiment, for each of the ten classes we conducted a varying number of subsearches. The linguistic cues for subsearches are given in Table 2.

To calculate similarity between images we used the correlation coefficient between grayscale values of the original images. The reason for this is that we performed classification based on gray-SIFT and CyColor features [27] and we did not want to use the same features for database cleaning and classification

Table 2: Linguistic cues for Google subsearches used for classification.

Basic search term	Meanings	Linguistic cues
apple	brand “Apple”, apple fruit	delicious, green, red ripe, unripe
cup	drinking cup, world cup, bra cup	coffee, empty, full, porcelain, tea
glass	drinking glass, vision-aid, glass a material	drinking, empty, full, juice, wine
kiwi	kiwi fruit, kiwi bird	fresh, fruit, green, juicy, ripe
oil	oil plant, cooking oil	cooking, food, olive, salad, sunflower
orange	orange fruit, brand “Orange”, orange color	fresh, fruit, juicy, ripe, sweet
peach	peach fruit, princess Peach	fresh, fruit, red, ripe
pot	cooking pot, flower pot, plant	aluminium, boiling, cooking, food, kitchen, metal
salmon	salmon fish, salmon dish	baked, cooked, marinated, salted, smoked, steamed
sponge	cleaning sponge, SpongeBob	cleaning, foam, household, kitchen, scrubbing

in order to avoid bias in evaluation process.

For this, we converted original images to grayscale images and resampled them to $100 \times 100\,px$. Finally, we calculated the distance d between image X and Y as follows:

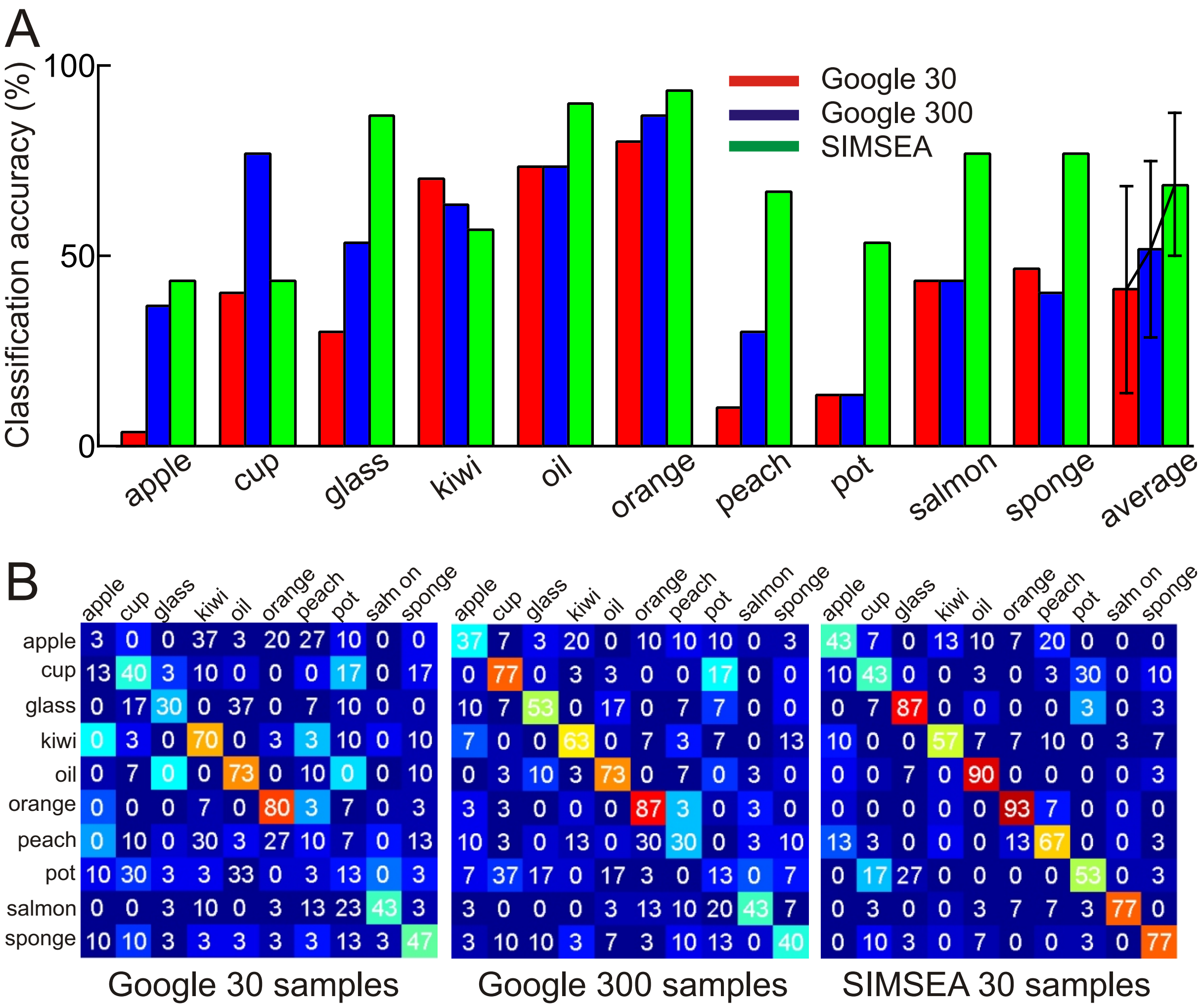
$$d(X,Y) = 1 - CC(X,Y), \tag{4}$$

where CC denotes the correlation coefficient between two images X and Y. We included an image into result set if $d < \theta$, where in this case we set $\theta = 0.1$. We used 300 images from the top of each sub-search in order to generate cleaned databases by SIMSEA.

For classification we generated three training sets: 1) first (from the top) 30 samples returned by Google search (Google 30), 2) first 300 samples returned by Google search (Google 300), and first (according to ranking r) 30 samples obtained by SIMSEA algorithm. For testing we generated a test set of 30 samples per class obtained by performing standard Google searches using queries from different (non-english) languages in order to avoid overlapping sets. Note that for the test set we manually selected only those images which were not present in training data sets. Training and test sets can be downloaded at http://www.dpi.physik.uni-goettingen.de/cns/index.php?page=simsea-benchmark.

We used the classification pipeline proposed by Schoeler et al. [27] which uses a combination of gray-SIFT and CyColor features. Local descriptors where extracted on a dense grid within the full image and oriented along the local image gradient. We compared performance of SIMSEA and Google search by looking at the classification accuracy. We expected that classification accuracy when using SIMSEA's training set will outperform those of Google training sets due to cleaner (with respect to the context) image sets.

3.2.2 Results



Google 30 samples

	apple	cup	glass	kiwi	oil	orange	peach	pot	sam on	sponge
apple	3	0	0	37	3	20	27	10	0	0
cup	13	40	3	10	0	0	0	17	0	17
glass	0	17	30	0	37	0	7	10	0	0
kiwi	0	3	0	70	0	3	3	10	0	10
oil	0	7	0	0	73	0	10	0	0	10
orange	0	0	0	7	0	80	3	7	0	3
peach	0	10	0	30	3	27	10	7	0	13
pot	10	30	3	3	33	0	3	13	0	3
salmon	0	0	3	10	0	3	13	23	43	3
sponge	10	10	3	3	3	3	3	13	3	47

Google 300 samples

	apple	cup	glass	kiwi	oil	orange	peach	pot	salmon	sponge
apple	37	7	3	20	0	10	10	10	0	3
cup	0	77	0	3	3	0	0	17	0	0
glass	10	7	53	0	17	0	7	7	0	0
kiwi	7	0	0	63	0	7	3	7	0	13
oil	0	3	10	3	73	0	7	0	3	0
orange	3	3	0	0	0	87	3	0	3	0
peach	10	3	0	13	0	30	30	0	3	10
pot	7	37	17	0	17	3	0	13	0	7
salmon	3	0	0	0	3	13	10	20	43	7
sponge	3	10	10	3	7	3	10	13	0	40

SIMSEA 30 samples

	apple	cup	glass	kiwi	oil	orange	peach	pot	sam on	sponge
apple	43	7	0	13	10	7	20	0	0	0
cup	10	43	0	0	3	0	3	30	0	10
glass	0	7	87	0	0	0	0	3	0	3
kiwi	10	0	0	57	7	7	10	0	3	7
oil	0	0	7	0	90	0	0	0	0	3
orange	0	0	0	0	0	93	7	0	0	0
peach	13	3	0	0	0	13	67	0	0	3
pot	0	17	27	0	0	0	0	53	0	3
salmon	0	3	0	0	3	7	7	3	77	0
sponge	0	10	3	0	7	0	0	3	0	77

Figure 5: Classification results. **A)** Comparison of classification accuracy for different training data sets: Google 30 samples, Google 300 samples and SIMSEA 30 samples. B) Confusion matrices. Numbers correspond to classification accuracy (%).

The results of classification experiment are presented in Fig. 5 where summarized results are shown in panel A and confusion matrices for each method are given in panel B. First of all (see panel A), we observe that Google 300 gave better classification accuracy on average as compared to Google 30 (51.66% to 41.00%), since bigger training set (300 samples) includes relevant as well as irrelevant images, whereas first 30 images returned by Google search many times can mainly consist of irrelevant images, e.g., see Fig. 6. Classification accuracy when SIMSEA's training set was 68.66%. We obtained 27.66% of improvement in classification accuracy compared to Google 30 (for individual classes paired T-test returned score $p = 0.0038$) and 17% of improvement as compared to Google 300 (paired T-test score $p = 0.0511$).

To visualize performance of SIMSEA algorithm we show images for selected three classes (apple, oil and pot) returned by Google search and SIMSEA algorithm in Fig. 6. The first ten images returned by Google search are shown in upper row whereas the first ten (according to ranking r) images obtained by SIMSEA algorithm are shown in bottom row. We can see that in all three cases Google search results include images of classes from domains others then the desired kitchen domain. In contrast, SIMSEA was successful in eliminating these (context) irrelevant images.

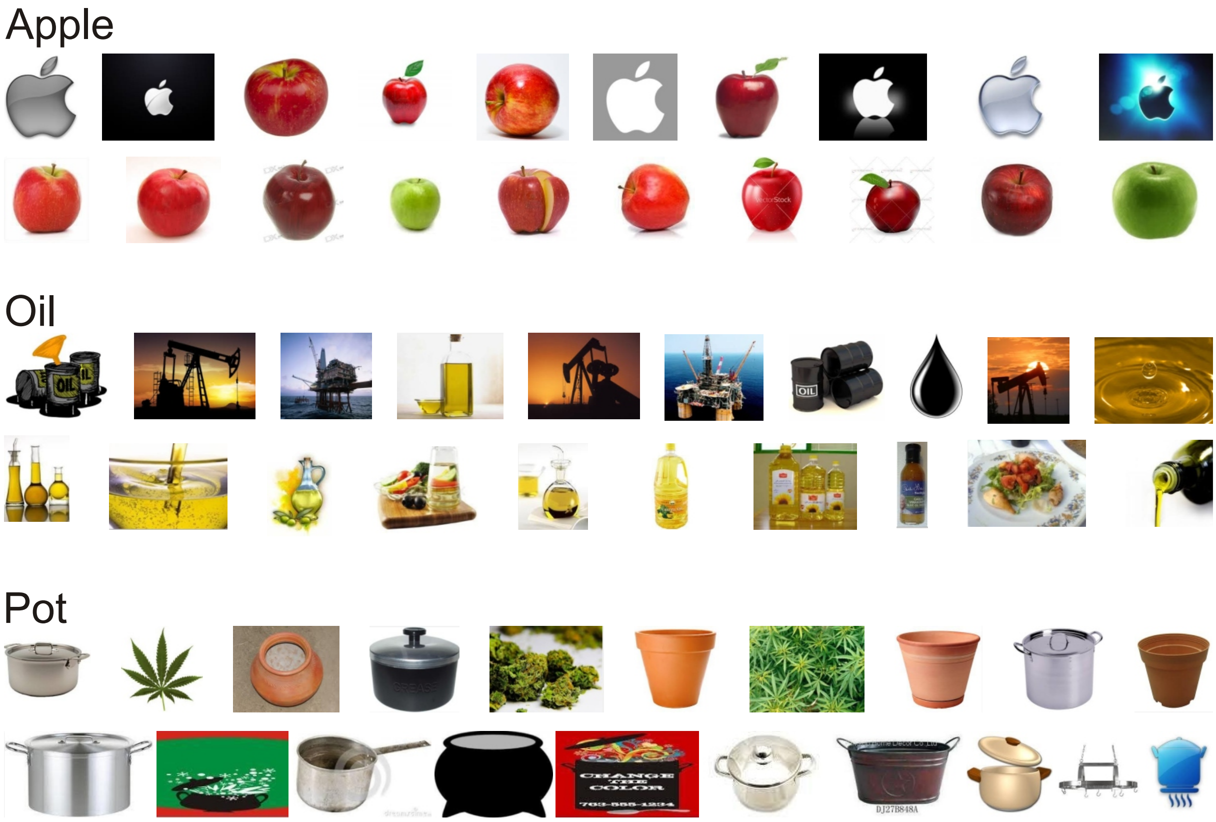


Figure 6: Images obtained by Google search and SIMSEA algorithm for three different searches:"apple", "oil" and "pot". Here we show first ten images returned by Google search (upper row) and first ten (according to ranking r) images returned by SIMSEA algorithm (bottom row).

4 Discussion

We proposed a method based on the combination of linguistic cues with the image domain that is useful for retrieving cleaner results in image searches, in particular it is able to tackle the problem of polysemes and homonyms. This is a novel approach and we have given the proof of principle by showing that it indeed leads to cleaner search results. The method is developed having autonomous robotic scenarios in mind, where robot on its own has to collect relevant images from internet, in order to execute instructions with objects he has not seen or been operating before.

One can ask where robots can obtain language labels and language cues from. Currently, the research in robotic systems performing human-robot interaction using natural language communication is quite advanced. [15, 4, 9]. It is desirable that in human environments robots communicate with humans in natural language. Thus robots would obtain language commands from humans, where not only objects and actions, but also context cues from natural language can be obtained. The other example of language-enabled robots are the robots executing natural language instruction sheets [31, 2]. The image database cleaning algorithm presented in this paper is developed having such robotic systems in mind.

Usually, for object recognition, the training data is gathered manually by a human [23, 24, 29]. The presented method allows (given a specific context) to gather training data automatically, thus, object learning/recognition can be done in an unsupervised manner, without human intervention, which is a big advantage in case of robot scenarios where one has to deal with many different objects. This is a common case in service robotics where robots need to operate in complex human environments.

Although we have introduced the notion of linguistic cues, we have not tackled the issue where these cues might come from, or how they should best be chosen. Automated extraction of object descriptors (cues) can be done using methods of natural language processing [8, 26, 22]. However, this is an issue falling in the domain of linguistics and is not the core of this paper.

Similar to the effectiveness of human linguistic refinement to distinguish intended meaning from other, our method has its strength when dealing with polysemes or homonyms. It is obvious that our method can only be as good as the subsearch results which depend on the "right" linguistic cues. If unrelated images occur in many of the subsearches, these images will erroneously be part of the result set.

In summary, we believe that this a novel and promising idea for data "cleaning" which can be used to automatically form training data sets using Internet search which later can be used for object classification/recognition and generalization. In future work we are going make such image search completely automatic by augmenting it with an automated extraction of object descriptors from language.

5 Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Programme and Theme: ICT-2011.2.1, Cognitive Systems and Robotics) under grant agreement no. 600578, ACAT.

References

[1] P. Perona A.D. Holub, P. Moreels. Unsupervised clustering for google searches of celebrity images. *8th IEEE Int. Conf. Automatic Face and Gesture Recognition*, 2008.

[2] Michael Beetz, Ulrich Klank, Ingo Kresse, Alexis Maldonado, Lorenz Mösenlechner, Dejan Pangercic, Thomas Rühr, and Moritz Tenorth. Robotic Roommates Making Pancakes. In *11th IEEE-RAS Int. Conf. on Humanoid Robots*, pages 529–536, Bled, Slovenia, October, 26–28 2011.

[3] Tamara L. Berg and David A. Forsyth. Animals on the web. In *CVPR*, pages 1463–1470, 2006.

[4] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus. Multi-view object recognition using view-point invariant shape relations and appearance information. In *International Symposium on Experimental Robotics (ISER)*, 2012.

[5] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *ACM Int. Conf. Image and Video Retrieval*, 2007.

[6] Anna Bosch, Andrew Zisserman, and Xavier Muoz. Image classification using random forests and ferns. In *ICCV*, pages 1–8, 2007.

[7] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30:107–117, April 1998.

[8] P. Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer Verlag, 2006.

[9] R. Deits, S. Tellex, P. Thaker, D. Simeonov, T. Kollar, and N. Roy. Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction*, 2(2):58–79, 2013.

[10] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *10th IEEE Int. Conf. Computer Vision*, volume 2, pages 1816–1823, oct. 2005.

[11] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, pages 264–271, 2003.

[12] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *8th Europ. Conf. Computer Vision*, pages 242–256, May 2004.

[13] Rick Grush. The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, 27:377442, 2004.

[14] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multimodal semi-supervised learning for image classification. In *CVPR*, 2010.

[15] H. Holzapfel, D. Neubig, and A. Waibel. A dialogue approach to learning object descriptions and semantic categories. *Robotics and Autonomous Systems*, 56(11):1004–1013, 2008. ¡ce:title¿Semantic Knowledge in Robotics¡/ce:title¿.

[16] Li jia Li, Gang Wang, and Li Fei-fei. Optimol: automatic online picture collection via incremental model learning. In *CVPR*, 2007.

[17] Yushi Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1877 –1890, Nov. 2008.

[18] Inayatullah Khan, Peter M. Roth, and Horst Bischof. Learning object detectors from weakly-labeled internet images. In *35th OAGM/AAPR Workshop*, 2011.

[19] Jens Kober, Andreas Wilhelm, Erhan Oztop, and Jan Peters. Reinforcement learning to adjust parametrized motor primitives to new situations. *Auton. Robots*, 33(4):361–379, 2012.

[20] K. Kronander, M.S.M. Khansari-Zadeh, and A. Billard. Learning to control planar hitting motions in a minigolf-like task. In *2011 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 710 –717, sept. 2011.

[21] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, Nov. 2004.

[22] J. J. McAuley, J. Leskovec, and D. Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *International Conference on Data Mining*, 2012.

[23] M. Muja, R. B. Rusu, G. Bradski, and D. G. Lowe. REIN-A fast, robust, scalable REcognition INfrastructure. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[24] Wail Mustafa, Nicolas Pugeault, and Norbert Krger. Multi-view object recognition using view-point invariant shape relations and appearance information. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[25] B. Nemec, R. Vuga, and A. Ude. Exploiting previous experience to constrain robot sensorimotor learning. In *11th IEEE-RAS Int. Conf. Humanoid Robots*, pages 727–732, oct. 2011.

[26] J. Olivie, C. Christianson, and J. McCarry. *Handbook of natural Language Processing and Machine Translation.* Springer, 2011.

[27] Markus Schoeler, Simon Christoph Stein, Alexey Abramov, Jeremie Papon, and Florentin Wörgötter. Fast self-supervised on-line training for object recognition specifically for robotic applications. In *VISAPP*, 2014.

[28] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *11th IEEE Int. Conf. on Computer Vision*, pages 1 –8, Oct. 2007.

[29] Yuyin Sun, Liefeng Bo, and Dieter Fox. Attribute based object identification. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[30] M. Tamosiunaite, I. Markelic, T. Kulvicius, and F. Worgotter. Generalizing objects by analyzing language. In *11th IEEE-RAS Int. Conf. Humanoid Robots*, pages 557–563, oct. 2011.

[31] Moritz Tenorth, Ulrich Klank, Dejan Pangercic, and Michael Beetz. Web-enabled Robots – Robots that Use the Web as an Information Resource. *Rob. & Automat. Magazine*, 18(2):58–68, 2011.

[32] A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Trans. Rob.*, 26(5):800–815, oct. 2010.

[33] A. Vedaldi and B. Fulkerson. Vlfeat – an open and portable library of computer vision algorithms. In *18th annual ACM Int. Conf. Multimedia*, 2010.

[34] Gang Wang, D. Hoiem, and D. Forsyth. Building text features for object image classification. In *IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1367 –1374, Jun. 2009.

Semi-Supervised Learning of Action Ontology from Domain-Specific Corpora

Irena Markievicz[1], Daiva Vitkute-Adzgauskiene[1], Minija Tamosiunaite[2]

[1]Faculty of Informatics, Vytautas Magnus University, Kaunas, Lithuania
{i.markievicz, d.vitkute}@if.vdu.lt
[2]Bernstein Center for Computational Neuroscience, University of Gottingen
m.tamosiunaite@if.vdu.lt

Abstract. The paper presents research results, showing how unsupervised and supervised ontology learning methods can be combined in an action ontology building approach. A framework for action ontology building from domain-specific corpus texts is suggested, using different natural language processing techniques, such as collocation extraction, frequency lists, word space model, etc. The suggested framework employs additional knowledge sources of WordNet and VerbNet with structured linguistic and semantic information. Results from experiments with crawled chemical laboratory corpus texts are given.

Keywords: action ontology, semi-supervised ontology learning, natural language processing, corpus linguistics, domain-specific corpus

1 Introduction

Design and use of intelligent, knowledge-based systems requires an adequate domain model which is normally designed in the form of an ontology presenting main concepts and their associations necessary for reasoning purposes. For example, such an ontology applied in robotics activity scenarios allows to define the knowledge field of a robot aimed at carrying out tasks in a specific-domain, e.g. in a kitchen specific, or chemistry lab specific domains. Task-oriented ontologies are usually designed as action ontologies, with action verbs being their main concepts and, also, different elements describing the action environment (e.g. action objects, tools, location, time, etc.).

This paper deals, specifically, with construction issues of action ontologies, concentrating on automated ontology building methods, i.e. on so-called ontology learning methods.

The main classifying points for ontology learning approaches are: a) a priori knowledge at the input (texts, preprocessed texts, dictionaries, other ontologies, etc.); b) learning methods (statistic vs. logical, etc.) [1]. Based on the scope of a priori knowledge, unsupervised and supervised ontology learning methods are defined. Unsupervised ontology learning is based on concept and association extraction from domain-specific texts, often containing some basic linguistic annotations (e.g. mor-

adfa, p. 1, 2011.

phological annotations, dependency parses). Supervised ontology learning assumes the use of supplementary labeled information (e.g. specifically annotated training corpora), structured semantic information (e.g. taxonomies, ontologies) and regular-expression based lexical patterns used for concept and association extraction.

This paper presents a semi-supervised method for action ontology building, using both unsupervised information extraction from domain-specific corpora, and, also, input from other ontologies or external databases with structured semantic information as well as corresponding lexical patterns for information extraction.

Experimental investigation is based on building of an action ontology for a robotics scenario, using a domain-specific corpus with crawled online material on chemistry laboratory processes. The corpus texts describe chemistry laboratory experiments, basic rules, instruments and techniques. The overall size of the experimental chemistry lab corpus (further referred to as the CHEMLAB corpus) is 1,971,415 running words. Collected texts were morphological annotated and lemmatized using Stanford University NLP tools for English language (http://nlp.stanford.edu/software/).

2 Related works

Related works can be grouped into those dealing with action ontology construction, and those dealing with automation of general domain ontology building processes.

Research works on action ontologies are in most cases oriented towards the development of domain-specific ontology models (knowledge structure) and reasoning mechanisms. Research domains are usually related either to natural language interfaces to agent systems [2,3], or structures for organizing work in robot-based systems [4,5]. However, little or no attention in these cases is paid to the automation of ontology creation process, with manual procedures prevailing, e.g. using ethnographic methods, study of human behavior and work practice [4]. Individual attempts of automated design of knowledge bases for understanding user situations and actions are usually rather limited to a priori knowledge structure, e.g. using semi-structured instruction texts [6].

References on automation of general domain ontology building process cover different design methods are much more, mainly based on transformations and merging of other existing ontologies, on domain text mining and use of external knowledge resources. [7] and [8] give a good summary of available automatic and semi-automatic ontology extraction techniques. Approaches using external knowledge resources, mainly WordNet [9] and those making use of different Natural Language Processing (NLP) methods [10] are prevailing. Semi-supervised methods, combining concept mining in domain texts and relationship extraction from WordNet are also presented in some works [11].

Our difference is in offering a semi-supervised ontology building method, specifically tailored for a domain-based action ontology design. It is based on text mining and NLP methods, combined with automated information extraction from several external knowledge bases – WordNet and VerbNet.

3 Action ontology learning model

General methodology for ontology building from texts can be described using the following meta-model [1]:

$$M = \{D, LA, T, S, C, TR\}, \quad (1)$$

where D is document collection (text corpus), LA are linguistic annotations for corpus texts, T is terminology collection, S is synonym collection, C is ontology concept collection, TR are ontology relations (associations).

Domain corpus texts, possibly with linguistic annotations, are used as the input to different NLP tools, resulting in terminology collection, further grouped into synonym collection (synsets). These are further used as building blocks for ontology concept collection, and the latter is finally enriched with corresponding associations between concepts.

There are different ontology development methodologies available. However, for building ontology from scratch using domain-specific corpus texts and integrating other knowledge sources, ontology engineering methodology named *Methontology* [12] is the most appropriate, as it suggests a framework for cyclical, multi-step ontology building, i.e. “ontology growing” based on the use of evolving ontology prototypes. For each prototype, *Methontology* suggests to start from planning, i.e. determining the time and resources necessary for each ontology building task. Ontology building starts from ontology specification, giving the domain, purpose, scope, knowledge source information. Further ontology development tasks include conceptualization (building a conceptual model), formalization (specifying techniques and tools) and implementation. Ontology development is accompanied by parallel activities of knowledge acquisition – extraction, integration, evaluation. Integration with other ontologies or knowledge databases should be described before implementation starts. Also, the evaluation of outcomes is foreseen by planning of control and quality assurance processes.

Further, the application of this model to the automated design of an action ontology for a robotics scenario is presented.

Each robot has limited number of actions, which it can execute. The action ontology should be based on those actions and it should add related actions and action environment information in the process of ontology *growing*. Knowledge sources, that are relevant in this case, consist of a domain-specific text corpus (chemistry laboratory domain is considered) and other related ontologies and other sources with structured semantic information. Linguistic database of English language WordNet (http://wordnet.princeton.edu/) and domain-independent verb lexicon for English language VerbNet (http://verbs.colorado.edu/verb-index/) were selected as the most appropriate external knowledge sources for action ontology building.

A conceptual model of the action ontology for a robotics scenario is given in Fig.1.

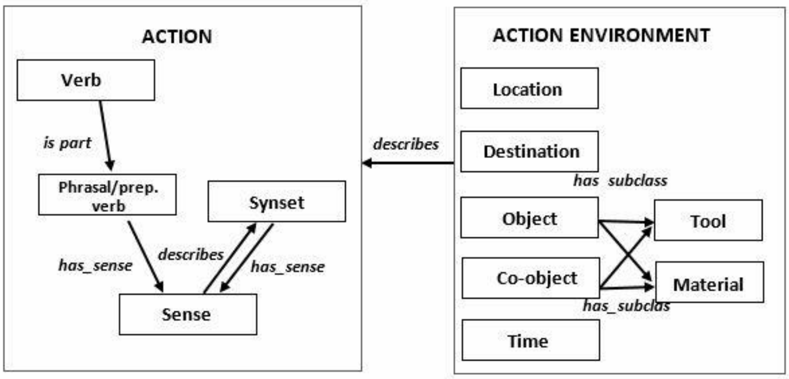


Fig. 1. Conceptual model of an action ontology

The presented action ontology conceptual model assigns appropriate action *synset* for each action and, also, all action details required for action execution (action environment). Action *synset* contains verbs, prepositional verbs and phrasal verbs, having the same sense. Environment description includes all the necessary elements for robot activity: time, location, destination, involved tools, involved material, etc.

Fig.2 presents the general process-structure of the semi-supervised action ontology building approach. Action verbs, extracted from morphological annotated corpus, are grouped into action synsets. In this process, external lexical data sources of WordNet and VerbNet databases are involved. These databases are also employed in action environment building. The elements in action environment synsets are grouped by the semantic roles, indicated by VerbNet frames and WordNet relations. Relations and axioms between ontology elements include semantic event chains and manually-built rules.

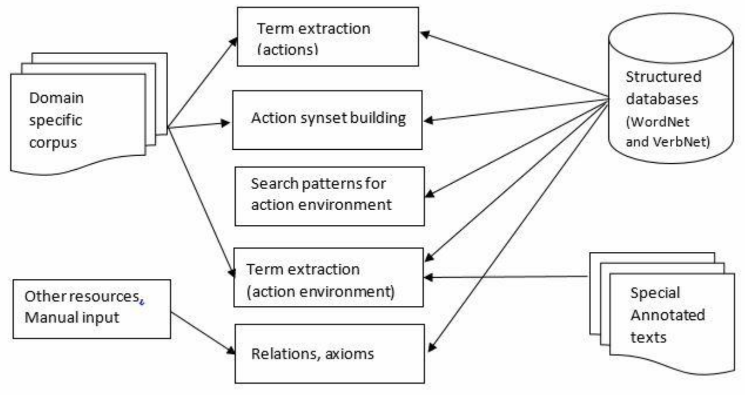

Domain specific corpus
Term extraction (actions)
Action synset building
Search patterns for action environment
Term extraction (action environment)
Relations, axioms
Other resources, Manual input
Structured databases (WordNet and VerbNet)
Special Annotated texts

Fig. 2. General process-structure of the semi-supervised action ontology building

The following sections give a step-by-step presentation of the action ontology building process, illustrated by experimental examples from the chemistry laboratory domain.

4 Term extraction (actions)

Extraction of terms denoting actions is the first step in the action ontology implementation process. With domain-specific corpus available, the most reasonable way to start building of the glossary of the most common actions is by building a verb-frequency list and filtering out the most frequent actions. In order to have a complete action representation, term-specific linguistic patterns, which include verbs, prepositional verbs (verb + preposition) and phrasal verbs (verb + [direct object] + adverb) and other multiword verbs (verb + direct object, verb + modifier) are used. Results of an experiment with the chemistry laboratory corpus by applying the above mentioned patterns on a morphologically annotated corpus is presented in Table 1.

Table 1. Actions by frequency - examples on MIX and PUT action groups

PUT	Frequency	MIX	Frequency
put in	353	mix with	342
put on	149	mix of	189
put into	111	mix to	178
put of	94	mix together	99
put away	43	mix for	53
put back	32	mix up	45
put to	29	mix into	30
put together	18	mix at	24
put not (n’t)	17	mix as	23
put off	16	mix until	21
put out	15	mix not (n’t)	17
put at	12	mix under	16
put down	11	mix by	13

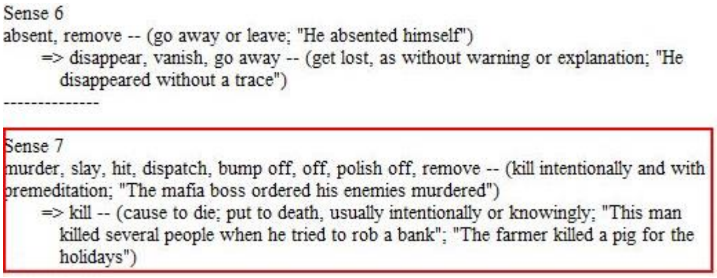
Larger frequency values point to the importance of an action verbs. When planning the ontology building process, these actions should be taken care first of all. Also, the experiment results point to the need of sorting out action verbs into synonymic groups, synsets, as actions linked to the same main verb can have entirely different meanings (e.g. “*put on*” and “*put off*”).

5 Building synsets of similar actions

A verb usually has more than one sense and its' sense can change in collocation with other words, e.g. a direct object name, a preposition or a certain modifier (e.g. *don't*). Data from Table 1 contains examples of verbs with similar meaning, which can be marked as synonyms: *put in = put into, put out = put away*. It also contains verbs with opposite meaning: *put in ≠ put off, put in ≠ put out, put ≠ put not (n't), mix ≠ mix not (n't)*.

Grouping actions into the synsets with the same sense is the next step of action ontology learning. This process involves external domain-independent lexical databases – *WordNet* and *VerbNet*. *WordNet* contains English language nouns, verbs, adjectives and adverbs. It describes the following relations between words: for nouns – hypernyms, hyponyms, holonyms and meronyms, for verbs – hypernyms, troponyms, for adjectives – relativeness, similarity, participation, for adverbs – common adjectival core. *VerbNet* groups English language verbs into conceptual classes. Each verb is described by roles and restrictions, its semantic group, frames with common examples and syntactic structure.

Synset is a synonym ring, which groups semantically equivalent data elements. Fig.3 presents an excerpt from synsets of verb "*remove*", as given by WordNet. Not all of them are adequate to the domain-specific action ontology – for example, *Sense-7* verbs, describing murder, are not adequate to the CHEMLAB domain.



```
Sense 6
absent, remove -- (go away or leave; "He absented himself")
      => disappear, vanish, go away -- (get lost, as without warning or explanation; "He
         disappeared without a trace")
--------------

Sense 7
murder, slay, hit, dispatch, bump off, off, polish off, remove -- (kill intentionally and with
premeditation; "The mafia boss ordered his enemies murdered")
      => kill -- (cause to die; put to death, usually intentionally or knowingly; "This man
         killed several people when he tried to rob a bank"; "The farmer killed a pig for the
         holidays")
```

Fig. 3. Synsets for verb "*remove*" (Source: WordNet)

Similar situation can be observed in VerbNet – verb "*remove"* is assigned to a semantic group, containing not just common synonym verbs (*extract*, *delete*, *dismiss*, *separate*, *etc.*), but also more specialized ones (*excommunicate*, *ostracize*) with their meaning dependent on the domain context.

Therefore, the task is to filter out inadequate verb senses and to grow synsets by adding suitable verbs with the same sense, coming from different sources. Word Space Model (WSM), which is based on the hypothesis that words with similar meanings will occur with similar neighbors, if enough text material is available [13], is used for testing *semantic similarity* of verbs. WSM is implemented by calculating feature vectors (frequency of co-occurrence with other words) for each word and measuring the distance between corresponding vectors. The feature vector of a certain verb is calculated, taking every occurrence of this verb in corpus texts, identifying meaningful words in the sentence-wise neighborhood of each occurrence, and building a vector with calculated measures of association between the verb and each of its neighborhood words. *Pointwise mutual information (PMI)* coefficient, describing relationship between the probability of the co-occurrence of two words and their individual distributions, is normally used as a probabilistic association measure in building such feature vectors:

$$PMI(A,B) = \log\frac{p(A,B)}{p(A)p(B)} = \log\frac{p(A|B)}{p(A)} = \log\frac{p(B|A)}{p(B)}, \tag{2}$$

where *p(A,B)* is the probability of A and B occurring together in the same context and *p(A), p(B)* – probabilities of their individual occurrence.

Feature vectors are then compared between each other using the cosine similarity method:

$$\cos(\theta) = \frac{A\cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}, \tag{3}$$

where *A* and *B* are feature vectors of verbs that are being compared.

Cosine similarity ranges from -1 to 1, where -1 means exactly opposite sense, 0 means independence, and 1 shows strong synonyms.

Table 2 presents an excerpt of feature vectors for verbs "*wash*" and "*rinse*" built using the CHEMLAB corpus as a reference.

Table 2. Excerpt of feature vectors for „*wash*“ and „*rinse*“

WORD	PMI (wash)	PMI (rinse)	WORD	PMI (wash)	PMI (rinse)
acetone	6,11	7,329	NaOH	4,99	5,756
Acid	6,15	4,108	Precipitant	7,32	6,071
careful	7,204	7,374	Product	4,19	4,276
Dilute	6,55	5,636	Residue	5,81	7,182
discard	11,749	8,58	Sodium	5,45	5,705
distilled	6,981	6,213	Solvent	4,96	3,289
Fume	8,387	8,156	Buret	0,00	8,077
funnel	5,74	4,66	Cake	9,32	0,00
addition	0,00	4,053	Color	0,00	4,386

After applying the cosine similarity method we get:

$$\cos(\theta) = \frac{A\cdot B}{\|A\|\|B\|} = 0{,}772615 \tag{4}$$

As the obtained cosine similarity value is close to 1, we can state, that verbs "*wash*" and "*rinse*" are similar and can be included in the same synset.

By applying WSM consequently to verbs in WordNet (WN) sense descriptions and VerbNet (VN) class descriptions, we observe an ontology learning process, named as *synset growing*. Fig.4 illustrates the synset growing process for the verb "*add*".

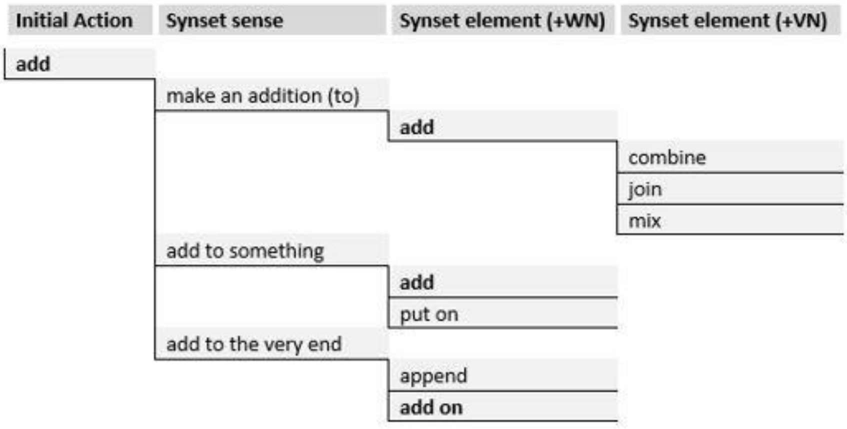


Fig. 4. Synset growing process – example for "*add*"

In this case, inadequate senses (e.g. "*add*" and "*add up*" in the meaning of "*summing up*") have been filtered as inadequate to the CHEMLAB domain.

6 Action environment learning

Each action synset should be described by a certain action environment, containing time, location, duration, destination, actor, tool, material, etc. elements. This process can be organized in 3 steps: 1) text preprocessing and building the glossary of possible environment elements; 2) obtaining rules (search patterns) for action environment element classification; 3) classifying the action environment elements by their roles.

Text preprocessing, leading to building of a glossary of possible action environment elements, involves collocation extraction methods. Collocation is a sequence of words that co-occur more often than it would be by chance (e.g. room temperature).

There are different statistical methods for extracting collocations from text, such Mutual Information, chi-squared test, Log-likelihood ratio, Fisher exact test, Dice coefficient, gravity counts [14], etc. Experiments showed, that for the purpose of identifying action environment elements, log Dice coefficient is adequate [14]:

$$logDice(A, B) = 14 + \log \frac{2|A \cap B|}{|A| + |B|}, \tag{5}$$

where |A ∩ B| is the frequency of A and B words co-occurrence in text, |A|, |B| - frequency of A and B words occurring separately.

Table 3 presents most frequent collocations obtained from the CHEMLAB corpus. Extracted glossary of CHEMLAB environment elements contains not just domain terms (e.g. *periodic table*), but also named entities, such as chemical elements (e.g. *carbon dioxide, etc.*), measurement data (e.g. *room temperature*) and names of tools (e.g. *water bath*).

Table 3. Most frequent CHEMLAB corpus collocations

Collocation	**logDice**	**Freq.**	**Collocation**	**logDice**	**Freq.**
reductive amination	13,740	287	aqueous layer	11,851	290
baking soda	13,709	206	science fair project	11,79	213
science fair	13,319	459	diethyl ether	11,784	232
carbon dioxide	13,098	361	reflux condenser	11,715	188
essential oil	12,891	296	acetic acid	11,696	426
periodic table	12,838	244	hydrochloric acid	11,579	359
copper sulfate	12,798	220	organic layer	11,482	202
hydrogen peroxide	12,705	270	small amount	11,404	207
methylene chloride	12,639	487	reaction mixture	11,337	787
sodium hydroxide	12,474	661	sassafras oil	11,222	284
reduced pressure	12,371	239	Chemical Abstracts	11,085	205
room temperature	12,359	551	sodium borohydride	10,872	196
alkali metal	12,285	200	formic acid	10,783	202
ammonium chloride	12,018	309	sodium acetate	10,766	213
sulfuric acid	11,856	504	sodium chloride	10,664	219

With action environment element glossary in place, classification of environment elements according to their action-specific roles must be done. This can be done by applying certain rules or search patterns. Possible sources for such rules may be the VerbNet lexicon with structured description of the syntactic behavior of verbs [15], or, alternatively, syntactic parse trees can be used. Our approach is based on automated extraction of rules from VerbNet lexicon database, mapping VerbNet thematic roles to the elements of the action environment conceptual model. Rules are extracted from VerbNet syntactic and semantic frames for corresponding verbs (Table 4).

Table 4. VerbNet syntactic and semantic frames for verb „*wash*“ (Source: VerbNet)

Description	**Syntax**	**Semantics**	**Example**

NP V NP	NP-**Agent**
	VB
	NP-**Object**
		TAKE CARE OF:
		ThemeRole = (?)Agent
		Event = during(E)
		ThemeRole = Object
			*He **washed** the **solvent layer**, dried it and concentrated.*

NP V	NP-**Agent**
	VB
		TAKE CARE OF:
		ThemeRole = Agent
		Event = during(E)
		ThemeRole = (?)Object
			***Wash** the **aqueous layer** twice.*

NP V NP
PP.instrument
	NP-**Agent**
	VB
	NP-**Object**
	PREP-**With**
	NP-**Instrument**
		TAKE CARE OF:
		ThemeRole = (?)Agent
		Event = during(E)
		ThemeRole = Object
		USE:
		ThemeRole = Agent
		Event = during(E)
		ThemeRole = Instrument
			*The filter cake is **washed** thoroughly with **methanol.***

NP V NP
PP.location
	NP-**Agent**
	VB
	NP-**Object**
	PREP-**In**
	NP-**Location**
		TAKE CARE OF:
		ThemeRole = (?)Agent
		Event = during(E)
		ThemeRole = (?)Object
		USE:
		ThemeRole = Agent
		Event = during(E)
		ThemeRole = Location
			*The top aqueous layer was **washed** in the **funnel.***

NP V NP
PP.duration
	NP-**Agent**
	VB
	NP-**Object**
	PREP-**During**
	NP-**Duration**
		TAKE CARE OF:
		ThemeRole = Agent
		Event = during(E)
		ThemeRole = (?)Object
		USE:
		ThemeRole = Agent
		Event = during(E)
		ThemeRole = Duration
			*The successive **washes** during the **work** up.*

In the example with "*wash*" verb, we obtain 5 possible search patterns, which are then used in action environment classification: *NP-Agent VB NP-Object; NP V; NP V NP PP.instrument; NP V NP PP.location; NP V NP PP.duration*. These patterns are then applied to morphologically annotated CHEMLAB corpus for filling the action ontology with classified action environment elements.

7 Experimental results

Experimental research with CHEMLAB domain corpus resulted in developing of a prototype action ontology containing 528 named classes, 3457 axioms (including 1070 logical axioms) and 1855 annotation assertion axioms. The following main classes were used for the action environment elements: ACTIVITY, OBJECT, CO-OBJECT, DESTINATION, TOOL, LOCATION and MATERIAL.

Ontology building process is illustrated for most common action verb from CHEMLAB domain corpus: add, apply, make, mix, pour, put, remove, transfer and wash. DL Expressivity is used for action ontology evaluation [16]. Developed ontology can be described with ALU metrics – allows atomic negation of concepts, that do not appear on the left hand side of axioms, concept intersection, concept union, universal restrictions [17].

Fig.5 presents the visualization of "*wash*" action and its environment. The "*wash*" synset in this case contains two synonyms: *wash* and *rinse.* Both actions can be directly connected with some objects: *filter*, *electrode*, *dish*, etc. Also, these actions are associated with other action environment elements: *duration*, *instrument* and *location*.

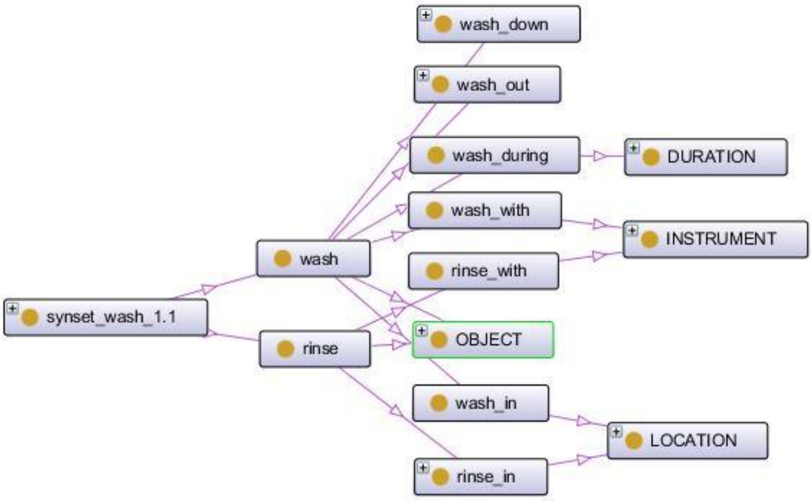
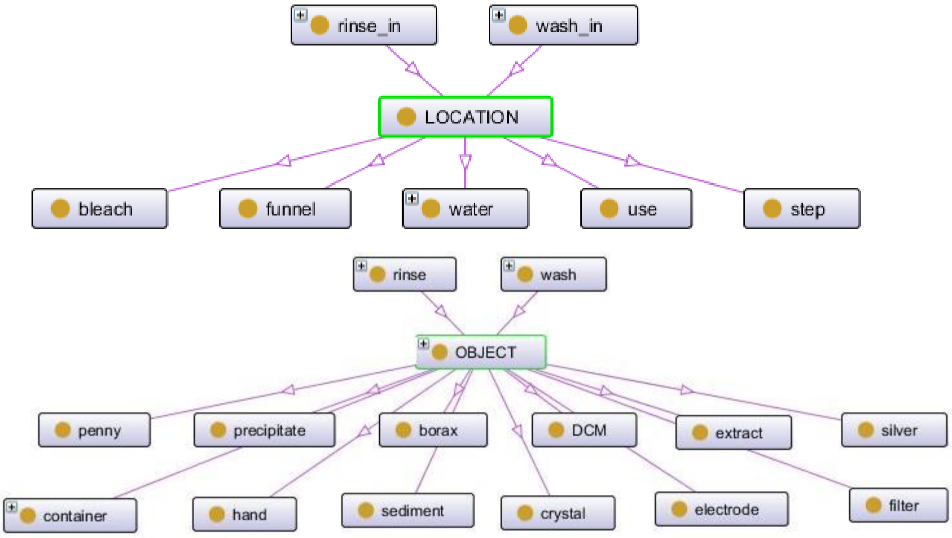


Fig. 5. "*Wash*" synset with its environment classes

Some segments of action environment are presented in Fig.6. The results of the experiments show, that the same element of environment can be defined as *location*, *object* or *instrument* depending on which preposition verb is used. E.g. *water* can be interpreted as *instrument* or *location*, depending on context, as shown in Fig 6.



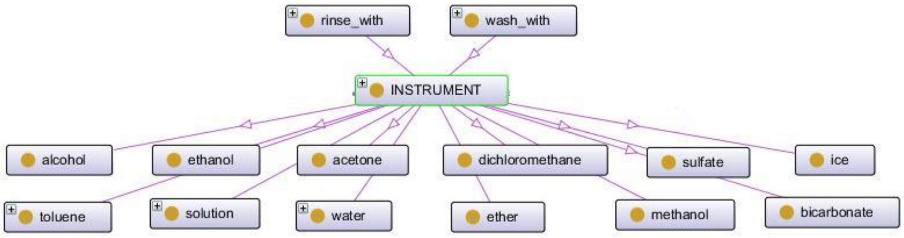


Fig. 6. Action environment examples for *location, instrument* and *object* in *wash* synset.

The elements of actions environment presented above were classified using VerbNet semantic frames. However, this method does not ensure, that all elements of action environment are classified. Different semantic roles of objects depend on action context. The results of the experiment show, that chemistry laboratory domain-corpus contains a lot of data, with multiple meanings and thus raises challenges for future work.

8 Conclusions

The proposed action ontology building approach uses employs NLP techniques: morphological analysis, POS tagging, collocation extraction, word space model for word sense disambiguation, concept classification and semantic tagging.

The results of this study show that structured information from existing knowledge bases (WordNet, VerbNet, etc.) can be of use in designing automated procedures both for ontology concept and relation learning.

A combination of unsupervised and supervised ontology learning methods is efficient for integrating different input data in action ontology building. This integration is specific to the each step of the proposed approach.

The designed prototype action ontology is still missing role hierarchy, inverse and functional properties. Adding cardinality restrictions would be helpful with chemical element measurement data.

More complex environment classification, recognition of hierarchical relations and building restrictions are the main tasks for future research work.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Programme and Theme: ICT-2011.2.1, Cognitive Systems and Robotics) under grant agreement no. 600578, ACAT.

References

1. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. pp. 10-17. Springer, Karlsruhe (2006)
2. Kemke, C.: An Action Ontology Framework for Natural Language Interfaces to Agent Systems. Artificial Intelligence Review, to be published (2009)
3. Morgenstern, L., Riecken, D.: SNAP: An Action-Based Ontology for E-commerce Reasoning. In: Formal Ontologies Meet Industry, Proceedings of the 1st Workshop "FOMI 2005" (2005)
4. Wales, R.C., Shalin, V.L., Bass, D.S.: Requesting Distant Robotic Action: An Ontology of Work, Naming and Action Identification for Planning on the Mars Exploration Rover Mission. Journal of the Association for Information Systems, vol. 8(2), art. 6. (2007)
5. Chatterjee, R., Matsuno, F.: Robot description ontology and disaster scene description ontology: analysis of necessity and scope in rescue infrastructure context. Advanced Robotics, vol. 19, no. 8, pp. 839-859. VSP and Robotics Society of Japan (2005)
6. Jung, Y., Ryu, J., Kim, K., Myaeng, S.: Automatic construction of large-scale situation ontology by mining how-to instructions from the web. Web Semantics: Science, Services and Agents on the World Wide Web 8(2), pp. 110–124 (2010)
7. Bedini, I., Nguyen, B.: Automatic Ontology Generation: State of the Art. PRiSM Laboratory Technical Report. University of Versailles (2007)
8. Grigonyte, G.: Building and Evaluating Domain Ontologies: NLP Contributions. Logos Verlag Berlin GmbH (2010)
9. Moldovan, D.I., and Girju, R.: Domain-specific knowledge acquisition and classification using WordNet. In: Proceedings of the Thirteenth International Florida Artificial Intelligence. Research Society Conference, pp. 224-228. AAAI Press (2000).
10. Nobécourt J.: A method to build formal ontologies from texts. In: Workshop on ontologies and text. Juan-Les-Pins, France (2000)
11. Hu, H., and Lium D.Y.: Learning OWL ontologies from free texts. In: Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference, vol. 2, pp. 1233-1237. IEEE (2004)
12. Mariano, F., Gomez-Perez, A., Juristo, N.: Methontology: From Ontological Art Towards Ontological Engineering. In: Proceedings of AAAI-97 Spring Symposium. Series on Ontological Engineering, pp. 33-40. AAAI Press, Stanford (2004)
13. Shutze, H., Pedersen, J.: A co-occurrence-based thesaurus and two applications to information retrieval. Information Processing and Management, 33(3). 307-318 (1997).
14. Daudaravicius, V., Marcinkeviciene, R.: Gravity counts for the boundaries of collocations. International Journal of Corpus Linguistics. 9(2), pp. 321-348. (2004)
15. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: Extending VerbNet with Novel Verb Classes. In: Fifth International Conference on Language Resources and Evaluation (LREC 2006). Genoa (2006)
16. Corcho, O., Fernández-López, M., & Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & knowledge engineering, 46(1), pp. 41-64. Elsevier (2003).
17. Baader, F.: Appendix: description logic terminology. The Description logic handbook: Theory, implementation, and applications, pp. 485-495. Cambridge University press (2003)

TRANSCLEAN: Unsupervised generation of training-sets for visual object recognition in robotics based on multi-language cues

Markus Schoeler, Florentin Wörgötter, Mohamad Javad Aein and Tomas Kulvicius
Bernstein Center for Computational Neuroscience (BCCN)
III. Physikalisches Institut - Biophysik, Georg-August University of Göttingen
{mschoeler, worgott, aein, tomas}@physik3.gwdg.de

Abstract— **Object recognition plays an important role in robotics, since object/tools first have to be identified in the scene before they can be manipulated/used. The performance of object recognition largely depends on the training dataset. Usually such training sets are gathered manually by a human operator, a tedious procedure, which ultimately limits the size of the dataset. One reason for manual selection of samples is that results returned by search engines often contain irrelevant images, mainly due to the problem of homographs (words spelled the same but with different meanings). In this paper we present an automated and unsupervised method, coined TRANSCLEAN, for generation of training sets which are able to deal with the problem of homographs. For disambiguation, it uses the context provided by a command like "tighten the nut" together with a combination of public image searches, text searches and translation services. We compare our approach against plain Google image search qualitatively as well as in a classification task and demonstrate that our method indeed leads to a task-relevant training set, which results in an improvement of 23% in object recognition for 15 ambiguous classes. In addition, we present an application of our method to a real robot scenario.**

I. INTRODUCTION

In the field of robotics object recognition plays an important role and is crucial for object manipulation tasks, since task specific objects/tools first have to be found and identified correctly before they can be used. To demonstrate, suppose we have a robot-scenario where we tell the robot to "fill the cup with water" as shown in Fig. 6. In order to recognize the bottle and the cup in the scene, the robot has to be trained on these objects beforehand. The training procedure is typically done by off-line training of a classifier with a pre-selected set of classes (images), where images are gathered manually by a human ([1], [2], [3], just to name a few), thus, in a supervised way. Some new approaches make use of Internet searches in order to get information about objects and instructions [4], [5], [6], [7]. Although modern search engines like Google or Yahoo can return a large number of images within milliseconds, not all of the returned images are task/context-relevant, especially due to the problem of homographs (polysemes), i.e., words that are spelled the same but which correspond to different meanings or objects. For example, the word "cup" can correspond to a cup for drinking, the world-cup or or bra's cup. "Apple" could mean the fruit, the brand logo or an Apple product. Nut could refer to a hex-nut or the food-nut (see Fig. 2 for an example).

In general, the performance of recognition systems heavily depends on the quality of the training data, thus, only task-relevant images should be collected. This is mostly done by searching for the class name in huge image databases (e.g., Google image search, Bing image search) and by selecting the most-relevant images. As this is especially non-trivial for search terms which are homographs, most recognition methods are trained using manually cleaned or even hand-made training-sets, the creation of which is a time consuming and tedious procedure. Moreover, if a certain task (like "tighten the nut") requires knowledge about an object which is not in the training set, execution is not possible and, even worse, new training images need to be taken or collected and cleaned manually before the robot is able to execute the task.

A lot of research exists on trying to solve this problem of dirty image search results, for example by making use of additional visual cues, e.g., local image patches, edges, texture, color, deformable shapes, just to name a few [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. All of these approaches use textual information, too. Either implicitly, by using the first results of text-based image search engines [9], [10], by constructing their own image search engine [12], [13], [18], or explicitly, by making use of image tags and labels as found in photo-sharing websites like Flickr [11], [12], [16]. To our knowledge, all of the above presented approaches achieve an improvement with respect to the quality of the result set. However, none of these methods can automatically cope with the problem of homographs (polysemes), which would be required in automated robotic applications like [4], [5], [6].

In this paper, in order to address the problem of homographs, we present a method for automatic (without human supervision) generation of task-relevant training sets for object recognition by using the information contained in a language-based command like "cut the apple" or "fill the cup". We ground our approach based on two facts: 1) homographs rarely occur for one word in multiple languages at the same time and 2) context information (action) provided by the command can be used in order to get rid of ambiguous and non-task relevant translations. In order to create such an automatic system we will employ a combination of publicly available image search engines, text search engines and

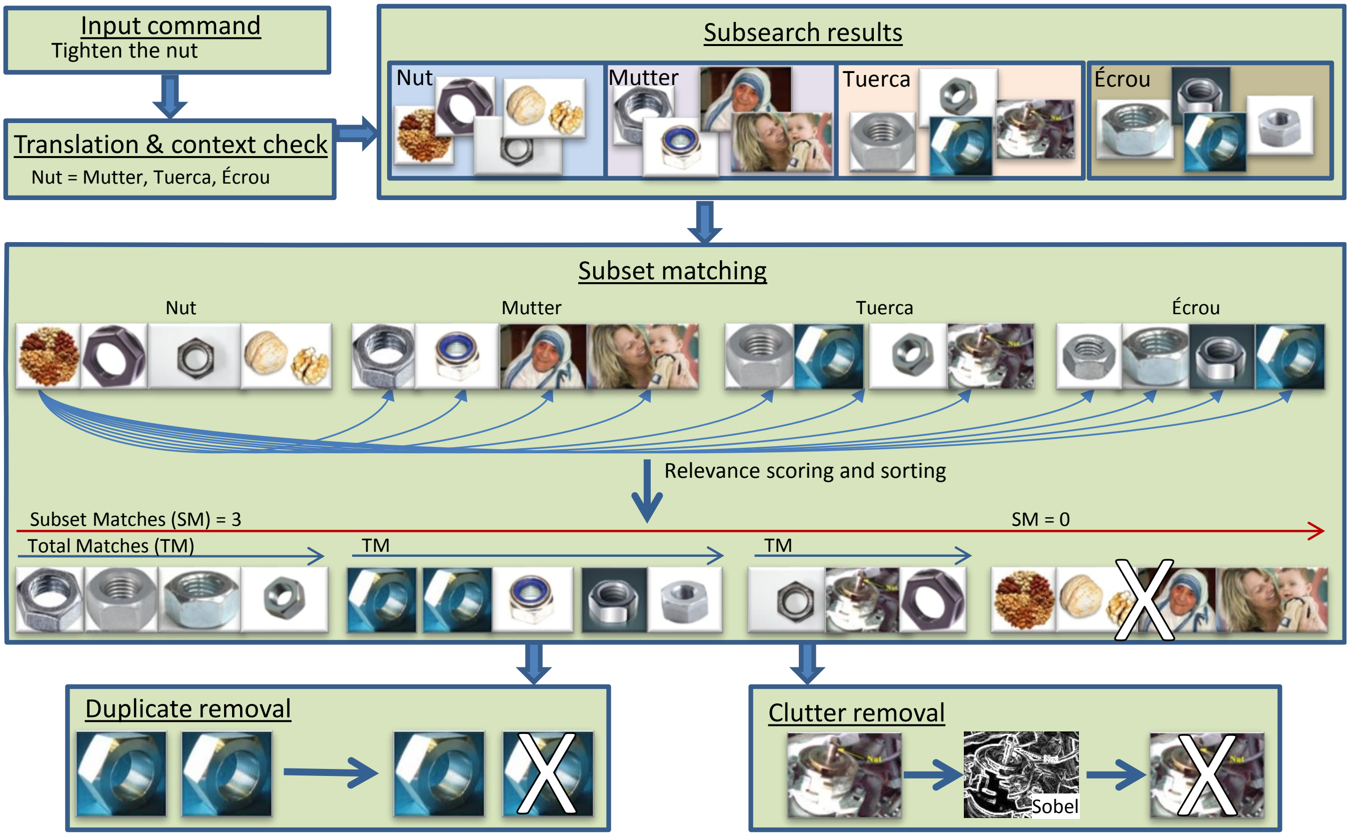


Fig. 1. Flow diagram of the TRANSCLEAN algorithm exemplified on the class "nut" in the context of "tighten". Subset Matches (SM) counts the total number of subsets in which a match has been found. Total Matches (TM) counts the total number of matches. SM is our first order and TM our second order relevance sorting criteria. Only images with SM $>$ 0 are considered further.

translation services.

The paper is organized as follows. First, we present our algorithm in detail in Section II. Then, in Section III, we show a qualitative comparison for selected classes (Section III-A) and evaluate the performance of our method quantitatively in an object classification task (Section III-B). Additionally we present an implementation of our method in a real-robot scenario (Section III-C). Finally, we conclude our study in Section IV by discussing our approach and comparing it to other existing methods.

II. THE TRANSCLEAN ALGORITHM

The algorithm consists of five sequential steps: 1) command translation, 2) context check, 3) image retrieval, 4) subset matching and 5) duplicate and clutter removal (see Fig. 1). In this section we will use the example of a robot which has no idea about the concept of a nut. Nut is a homograph and can mean either a hardware-nut or a food-nut. Generating a training set using a plain Google search for "nut" will not work. In this case even humans cannot infer which object nut refers to. However, the command "tighten the nut" or "crack the nut" provides valuable context information for disambiguation which we want the robot to leverage.

A. Command translation

The first step of the algorithm is to translate the command to different languages. For this we translated nouns and verbs separately. Note that in our study we used a fixed command syntax: verb/action + noun/object. A more general command syntax would require the usage of grammar analysis methods (i.e. parsers, [19]). In this paper we used four languages: English, German, Spanish, French and Portuguese. Here, Portuguese was only used in the case when translations into the other languages resulted in less than three different terms (e.g., orange is the same word in English, French and German). As an example we will show the generation of the German subset. The first three translations for nut, tighten and crack are shown in Table I. "Mutter" and "Schraubenmutter" correspond to the hardware-nut. "Nuss" corresponds to the food-nut. As one can see the double-meaning of nut is not present in German.

B. Context check

If the translation service returns more than one translation for the noun this step will perform a context check using Google text search. The idea here is that Google will return significantly less results for a phrase which does not make sense like "Nuss anziehen" (tighten the food-nut), compared to a reasonable phrase like "Mutter anziehen" (tighten the hardware-nut). To retrieve the right translation in the specific

TABLE I

First three English to German translations for nut, tighten and crack returned by www.dict.cc.

nut	tighten	crack
Nuss	anziehen	zerbrechen
Mutter	verschärfen	knacken
Schraubenmutter	straffen	zersplittern

context the algorithm uses the noun which gets most matches combined with any translation for the verb. Table II shows how the context relevant German translations for “nut” can be reliably determined. The relevant translations in German, French and Spanish for “crack the nut” are Nuss, Noix, Nuez. The translations for “Tighten the nut” are Mutter, Écrou and Tuerca.

TABLE II

Context check for “tighten the nut” and “crack the nut” using the number of exact matches returned by Google text search. The context relevant translation is marked bold.

“tighten the nut”		“crack the nut”	
Term	**Matches**	**Term**	**Matches**
Nuss anziehen	41	Nuss zerbrechen	7
Nuss verschärfen	0	**Nuss knacken**	**3.170**
Mutter anziehen	**2.160**	Mutter zerbrechen	30
Mutter verschärfen	5	Mutter knacken	17

C. *Image retrieval*

This step downloads images for all relevant translations. In the “tighten the nut” context it downloads images for Nut, Mutter, Écrou and Tuerca into 4 separate subsets. In the context of “crack the nut” it downloads images for Nut, Nuss, Noix and Nuez.

D. *Subset matching*

Task-relevant images can be found in all subsets, whereas images which correspond to irrelevant content can usually be found only in one set. Nut in the hardware context is a good example as it translates to the German word “Mutter” which is also a homograph meaning the hardware-nut as well as “mother” (see Fig. 2). While mother images are only found in the German and food-nut images only in the English subset, images of hardware-nuts are found in all subsets. For similarity matching we used the procedure proposed by Kulvicius et al. [7]. Here, different from Kulvicius et al., we used two relevance scores instead of one. The pseudo-code in Fig. 3 shows how the two scores are assigned to each image I_i^k : 1) the number of total matches TM_i^k as well as 2) the number of subsets where a match has been found SM_i^k. Only images which have a match in at least one other subset are considered, i.e., $SM_i^k > 0$. Images are then sorted in descending order first by the number of subset matches and second by the number of total matches found. This assures that most task-relevant images are found at the beginning of the list whereas borderline cases are found at the end.

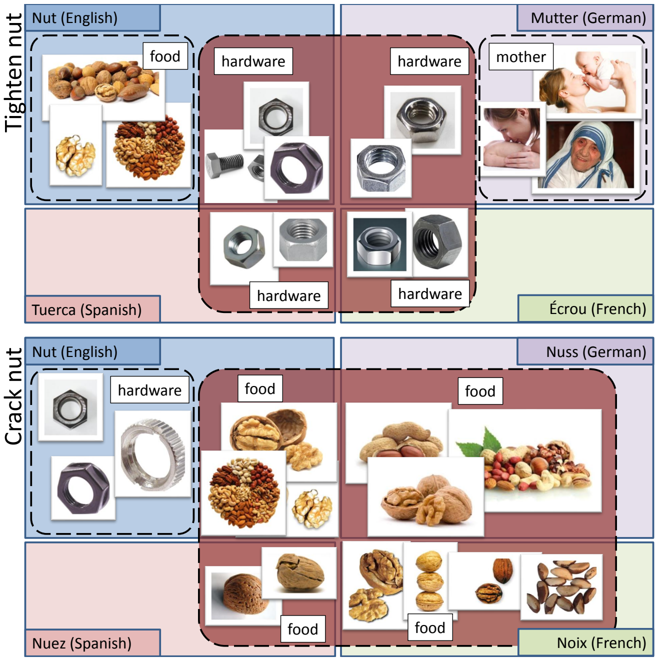


Fig. 2. Example word “nut” which is a homograph in English (food and hardware) and German (mother and hardware). By combining multiple languages and using the context check the proposed algorithm is able to retrieve the task relevant images for “nut” in both tasks (“tighten the nut” and “crack the nut”), the intersection marked with the red rectangle)

```
Get images $I_i^k$ ($k = 1 \dots m, i = 1 \dots n_k$), where
$m$ is the number of subsearches/languages considered and
$n_k$ is the number of images in subsearch $k$;
Set similarity threshold $\theta$;
Initialize total matches $TM_i^k = 0$.
Initialize subset matches $SM_i^k = 0$.
FOR $k = 1$ to $m$
    FOR $i = 1$ to $n_k$
        FOR $l = 1$ to $m$
            IF $k! = l$
                SubsetMatchFound = false
                FOR $j = 1$ to $n_l$
                    Compare images $I_i^k$ and $I_j^l$ by
                      calculating similarity $s$
                    IF $s > \theta$
                        increment($TM_i^k$);
                        IF not SubsetMatchFound
                            increment($SM_i^k$);
                            SubsetMatchFound = true;
```

Fig. 3. Pseudo-code for the subset matching to determine image relevance.

For similarity calculation the algorithm generates signatures using radially aligned gray-SIFT features as described in previous work [20] (the center is set to the middle of the image). Features are sampled on a dense grid on three scales. A bag-of-visual-words algorithm with 100 visual words is used to generate image signatures. As similarity measure we used the histogram intersection over all visual word bins. The similarity threshold θ was set to 0.7.

E. Duplicate and clutter removal

Finally, in order to clean the result set from duplicate images and images with cluttered scenes we perform a duplicate and clutter search. For that, we downscale all images to 100×100 pixels and generate gradient magnitude images g_i using the sobel filter. The similarity between image i and image j is calculated by L1-normalizing the gradient images g_i and g_j and calculating the histogram intersection. The duplicate threshold was fixed to 0.85 throughout all experiments. When a duplicate is found the image with lower relevance score is deleted. We additionally perform cluttered image removal by calculating the mean gradient magnitude within a five pixel image border. Using a clutter threshold of 0.1 effectively removed all images which were recorded in cluttered scenes and therefore considered bad for the training (e.g. an apple on a tree in a garden).

III. RESULTS

In order to evaluate the performance of the TRANSCLEAN algorithm we used 15 homographic classes. All classes, their possible meanings and action contexts are depicted in Table III. For classification we used the method proposed by [20] which uses a combination of gray-SIFT and CyColor features. Local descriptors are extracted on a dense grid and oriented along the dominant local gradient (the latter using the SURF detector). Three hundred visual words were used for the signature generation. A support-vector-machine with a histogram intersection kernel is used for the machine learning.

TABLE III

15 CLASSES USED IN OUR EXPERIMENTS. ALL TERMS HAVE MULTIPLE MEANINGS (NOT ALL ARE SHOWN). NAMES OF MOVIES, CHARACTERS AND BRANDS USUALLY DO NOT TRANSLATE INTO OTHER LANGUAGES. THIS IS WHY ACTION CONTEXT IS NOT ALWAYS REQUIRED. THE RELEVANT MEANING IS MARKED ITALIC. ONLY TRANSLATIONS AFTER CONTEXT CHECK ARE SHOWN.

Term	**Meanings**	**(Context:) Translations**
apple	*food*, laptop, logo	manzana, pomme, apfel
axe	*hardware*, brand	hacha, hache, axt
bolt	*hardware*, athlete, movie	tornillo, boulon, bolzen
cup	*drinking*, trophy, bra	**fill**: taza, tasse
fork	*hardware*, bike-part	tenedor, fourchette, gabel
glass	*drinking*, material	**fill**: vaso, verre
hammer	*hardware*, brand	martillo, marteau
nail	*hardware*, finger	**hit**: clavo, clou, nagel
nut	*hardware*, food	**tighten**: tuerca, ecrou, mutter
oil	*food*, mineral-oil	**eat**: aceite, huile, oel
orange	*food*, color	**cut**: laranja, naranja
pan	*hardware*, movie, god	sarten, poele, pfanne
peach	*food*, computer character	molocoton, peche, pfirsich
pot	*hardware*, drug	cacerola, casserole, topf
saw	*hardware*, movie	sierra, scie, saege

A. Qualitative comparison

To visualize the qualitative performance of the algorithm Fig. 4 shows the first 10 images retrieved by Google image search and by the TRANSCLEAN algorithm for three selected classes: axe, bolt and cup. The problem of homographs is especially obvious in the case of axe and bolt showing solely irrelevant content except one image. The Google search for cup results in roughly 50% unrelated images. In contrast, the TRANSCLEAN algorithm yields almost a 100% clean image set for all classes.

B. Image classification

Additionally we tested the performance of the TRANSCLEAN algorithm quantitatively in an image classification experiment. We wanted to prove that training a classifier with images obtained with TRANSCLEAN results in significantly better classification accuracy as compared to training with uncleaned Google images. For comparison we generated three training sets: two returned from Google search using the first 30 and 300 images (Google 30 and Google 300) per class and one obtained from TRANSCLEAN. For testing we manually created a disjoint set containing only task-relevant images obtained from Google searches using other languages. Fig. 5 shows the confusion matrices for all three training sets. We can observe that Google 300 yields better accuracy than the Google 30 set. The reason for this is the lack of relevant images in the Google 30 set for many classes (some classes showing only one or two relevant images). Using Google 300 more relevant images are used but the performance is still worse than TRANSCLEAN due to noise caused by irrelevant images, resulting in relatively large intra-class variance. The most difficult classes were bolt, peach and nail. Bolt, for instance, was difficult since it is not well defined, sometimes showing a screw and sometimes showing a pin in all languages. Peach was often confused with an orange as they are very similar in appearance. As expected, the TRANSCLEAN algorithm improved classification accuracy on average by 24% compared to Google 300, resulting in a 63% classification accuracy.

C. Robotic application

Last but not least, we applied our method to a real-robot application where we let a KUKA LWR robot-arm [21] perform three actions (see Fig. 6):

1) "fill the cup" (with water from a bottle)
2) "tighten the nut"
3) "crack the nut" (with a wooden cube)

For each action only one object is task relevant. In all cases the robot needs to ignore all distractors and choose the right object depending on the action context. Several aspects, like object recognition and robot movement execution, rely on published works and will not be described here in detail. To extract objects from the scene we used the object extraction pipeline of [20] using RGB-D data for segmentation and high resolution images (4928×3264 pixels) for object recognition. We also recorded a background class consisting of many

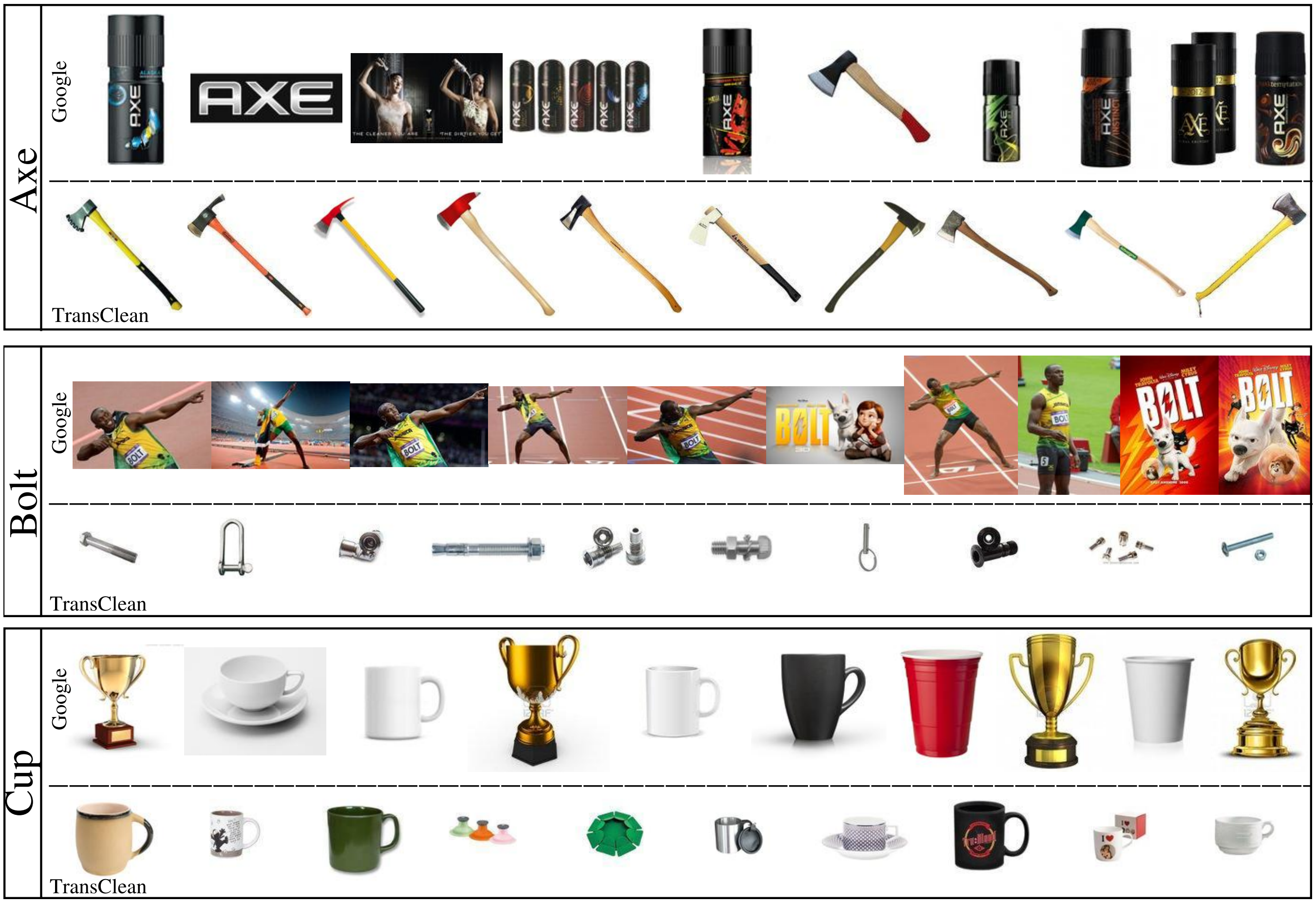
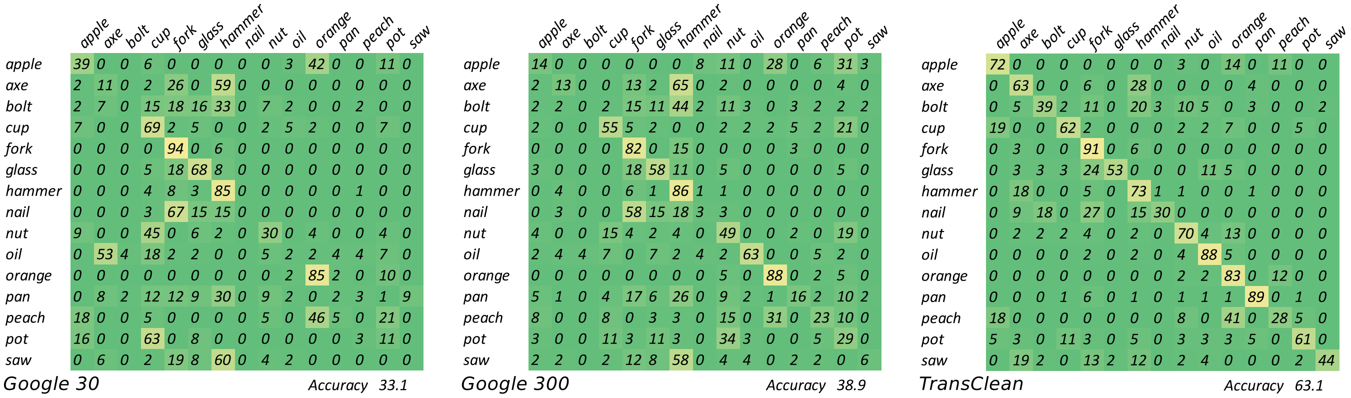


Fig. 4. Images retrieved by Google image search and by our TRANSCLEAN algorithm for three example classes. Only the first 10 Google images (top rows) as well as the 10 highest scoring TRANSCLEAN images (bottom rows) are shown.



Google 30 — Accuracy 33.1

	apple	axe	bolt	cup	fork	glass	hammer	nail	nut	oil	orange	pan	peach	pot	saw
apple	39	0	0	6	0	0	0	0	0	3	42	0	0	11	0
axe	2	11	0	2	26	0	59	0	0	0	0	0	0	0	0
bolt	2	7	0	15	18	16	33	0	7	2	0	0	2	0	0
cup	7	0	0	69	2	5	0	0	2	5	2	0	0	7	0
fork	0	0	0	0	94	0	6	0	0	0	0	0	0	0	0
glass	0	0	0	5	18	68	8	0	0	0	0	0	0	0	0
hammer	0	0	0	4	8	3	85	0	0	0	0	0	1	0	0
nail	0	0	0	3	67	15	15	0	0	0	0	0	0	0	0
nut	9	0	0	45	0	6	2	0	30	0	4	0	0	4	0
oil	0	53	4	18	2	2	0	0	5	2	2	4	4	7	0
orange	0	0	0	0	0	0	0	0	0	2	85	2	0	10	0
pan	0	8	2	12	12	9	30	0	9	2	0	2	3	1	9
peach	18	0	0	5	0	0	0	0	5	0	46	5	0	21	0
pot	16	0	0	63	0	8	0	0	0	0	0	0	3	11	0
saw	0	6	0	2	19	8	60	0	4	2	0	0	0	0	0

Google 300 — Accuracy 38.9

	apple	axe	bolt	cup	fork	glass	hammer	nail	nut	oil	orange	pan	peach	pot	saw
apple	14	0	0	0	0	0	0	8	11	0	28	0	6	31	3
axe	2	13	0	0	13	2	65	0	2	0	0	0	0	4	0
bolt	2	2	0	2	15	11	44	2	11	3	0	3	2	2	2
cup	2	0	0	55	5	2	0	0	2	2	2	5	2	21	0
fork	0	0	0	0	82	0	15	0	0	0	0	3	0	0	0
glass	3	0	0	0	18	58	11	0	5	0	0	0	0	5	0
hammer	0	4	0	0	6	1	86	1	1	0	0	0	0	0	0
nail	0	3	0	0	58	15	18	3	3	0	0	0	0	0	0
nut	4	0	0	15	4	2	4	0	49	0	0	2	0	19	0
oil	2	4	4	7	0	7	2	4	2	63	0	0	5	2	0
orange	0	0	0	0	0	0	0	0	5	0	88	0	2	5	0
pan	5	1	0	4	17	6	26	0	9	2	1	16	2	10	2
peach	8	0	0	8	0	3	3	0	15	0	31	0	23	10	0
pot	3	0	0	11	3	11	3	0	34	3	0	0	5	29	0
saw	2	2	0	2	12	8	58	0	4	4	0	2	2	0	6

TransClean — Accuracy 63.1

	apple	axe	bolt	cup	fork	glass	hammer	nail	nut	oil	orange	pan	peach	pot	saw
apple	72	0	0	0	0	0	0	0	3	0	14	0	11	0	0
axe	0	63	0	0	6	0	28	0	0	0	0	4	0	0	0
bolt	0	5	39	2	11	0	20	3	10	5	0	3	0	0	2
cup	19	0	0	62	2	0	0	0	2	2	7	0	0	5	0
fork	0	3	0	0	91	0	6	0	0	0	0	0	0	0	0
glass	0	3	3	3	24	53	0	0	0	11	5	0	0	0	0
hammer	0	18	0	0	5	0	73	1	1	0	0	1	0	0	0
nail	0	9	18	0	27	0	15	30	0	0	0	0	0	0	0
nut	0	2	2	2	4	0	2	0	70	4	13	0	0	0	0
oil	0	0	0	0	2	0	2	0	4	88	5	0	0	0	0
orange	0	0	0	0	0	0	0	0	2	2	83	0	12	0	0
pan	0	0	0	1	6	0	1	0	1	1	1	89	0	1	0
peach	18	0	0	0	0	0	0	0	8	0	41	0	28	5	0
pot	5	3	0	11	3	0	5	0	3	3	3	5	0	61	0
saw	0	19	2	0	13	2	12	0	2	4	0	0	0	2	44

Fig. 5. Confusion Matrix in percent. Rows: Actual class label. Columns: Predicted class labels.

random images taken from the scene with all distractors but without the hardware-nut, food-nut, trophy-cup and coffee-cup.

In case 1) the robot finds out that cup refers to the coffee-cup and ignores the trophy-cup. In case 2) the food-nut is ignored since the TRANSCLEAN algorithm only generates training images for the context relevant hardware-nut. Using the context "crack" in case 3) the robot finds the food-nut and ignores the hardware-nut. Note that in our case the commands were typed directly into the computer program. Additionally, we started with the bottle grasped by the robot hand and the location of the wooden-cube known to the robot beforehand. Consequently, the task for the robot was to find out and recognize which cup and nut the commands refer to and to execute the corresponding action.

For action execution we used the library of manipulation actions from [22], which is based on semantic event chains [23] and modified dynamic movement primitives [24]. Here,

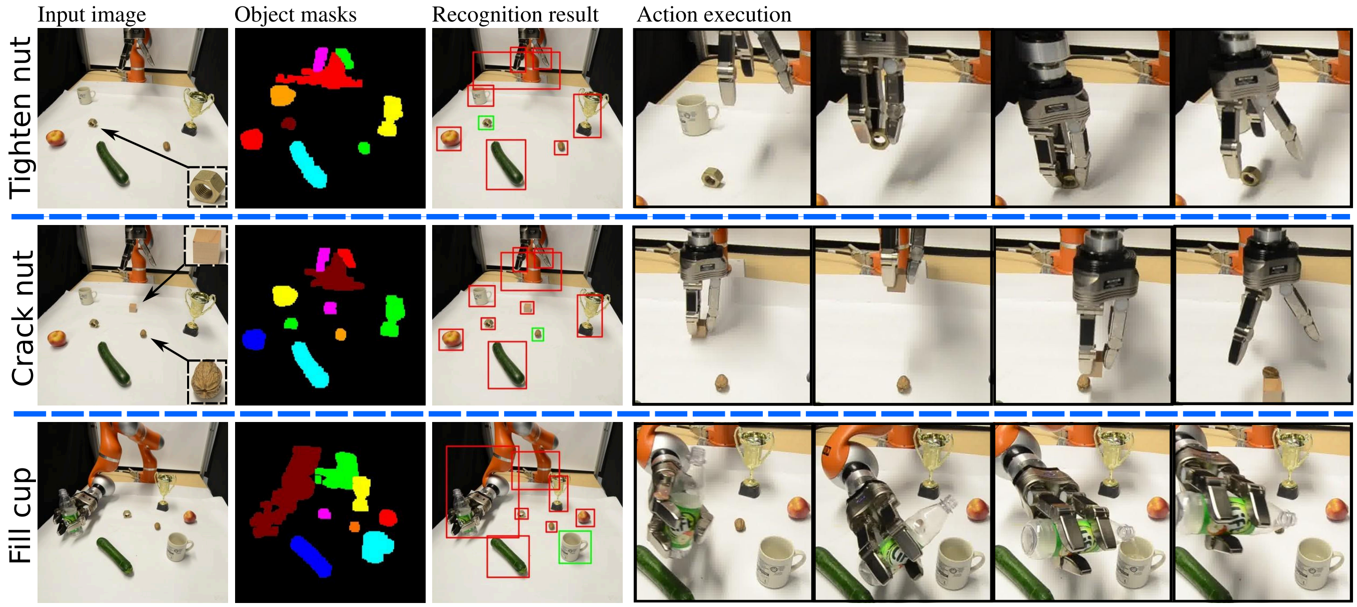


Fig. 6. Three example scenes where the robot had to perform the actions "tighten the nut", "crack the nut" and "fill the cup". The robot starts without knowledge about cups and nuts. In addition to the objects involved in the action we put other items as distractors into the scene. One of them being a different type of cup (the trophy) (see Fig. 4). Even though two items can be referred to by the word nut, only one of them is relevant for the specific action. The robot uses the TRANSCLEAN algorithm to determine the context relevant objects and generates a training set on-the-fly. RGB-D information is used to generate object masks and a high resolution image is used for the classification (see [20] for details). The green box marks the object which gets the highest score from the classifier.

specifically, we used pouring, picking-up and putting-down actions. Object positions came directly from the object extraction. The action "tighten" is a complex action sequence and consists of "pick up", "put on" and "turn". "Put on" and "turn" are difficult actions which require detailed knowledge about the objects. As this is not in the focus of this paper, we only required the robot to execute the first step of this action.

In Fig. 6 we demonstrate that the robot successfully recognized the cup for filling, the hardware-nut for tightening and the food-nut for cracking. Please also see the supplementary video of these experiments.

IV. DISCUSSION

In this paper we presented a method for automated generation of task-relevant training-sets for object recognition by combining image search engines, text search engines and translation services. The method is useful for obtaining "cleaner results" in image searches. While this is already a valuable property of the algorithm, it is of particular importance in the case of homographs. We showed that the presented approach indeed leads to cleaner search results and better recognition rates as compared to plain Google search. The method was developed with autonomous robotic systems in mind, that is where a robot has to collect (without human supervision) relevant images from the internet, in order to disambiguate and execute human instructions. In this section we will discuss our approach and how it relates to other existing methods.

In the field of artificial intelligence and computer vision object classification is considered one of the hardest tasks. Due to its importance for many applications, including robotic systems, a lot of effort has been made in order to improve the performance of recognition methods. Progress has been made in several aspects of recognition: 1) the description of visual information (e.g., using local descriptors like SIFT [25], SURF [26]), 2) the way local features are compressed to form short-as-possible, but still discriminative image signatures (e.g., Bag of Visual Words [27], Spatial Pyramids [28], Fisher Vectors [29], Pooling [30]) and 3) in the development of machine learning algorithms which can generalize from the training signatures to meaningful class concepts (e.g., Random Forests, Support-Vector-Machines, Neural Networks, [31], [32]). Additionally, as shown above, recognition performance highly depends on the quality of the training-set. Generating such training-sets for robotic applications by a human operator is a very time consuming and tedious procedure, which also limits the size of the training set. On the other hand, keeping only the first pages returned by Google [10] limits the size of the training set even more, and worse, will not work at all for homographic classes (see Fig. 4). The approach presented here provides a solution to solve such problems, based on the additional context information provided by the task (command) and four different subsearches (languages) to automatically retrieve clean training sets. Additionally, adding more languages/subsets (especially with different roots) and several search engines should lead to larger datasets, a more fine-grained relevance score and therefore an even greater improvement in object recognition performance. One could also improve the results by using state-of-the-art image retrieval algorithms like [33] for the image matching and duplicate removal.

We have shown that our method performs well as long as the actions allow inference of context, as with fill, crush, crack, pour, cut, screw on, tighten, nail down, and so on. However, performance will drop if actions are used which can be applied to many objects in different contexts, as is generally the case with actions like give, put, move, place, lift and throw. Nevertheless, even humans would experience this problem and would require additional information (if the context is not known beforehand) in cases like "give me the nut".

Our approach most closely relates to the approaches of Kulvicius et al. [7] and Tamosiunaite et al. [6]. In [7] additional language cues are used in order to perform several sub-searches based on specific context. For example, to generate a task-relevant dataset for the class cup it could use "coffee cup", "tea cup", "full cup", "empty cup", etc. Such context-dependent cues can be obtained from language analysis. However, this requires knowledge about the domain as well as collecting a text-corpora for each specific context. In contrast, in the current approach there is no need for such information, and the context, if needed (in case multiple possible translations exist), is provided by the action (verb). Similar to our approach, Tamosiunaite et al. [6] make use of language and actions together with Google text search in order to boot-strap in the object domain and to find out which other objects could be used as a replacement.

If the command is "cut the cucumber", then the algorithm would return that carrots, potatoes, apples, etc. can be cut, too. Different from [6], we use the action for a different purpose, i.e., in order to select the relevant context.

As explained above, our approach requires textual (language-based) cues in order to perform image searches. In our study these cues were entered manually in a computer program as a text-command. However, such cues could come from human-robot interaction using natural language communication [34], [35], [36]. Thus robots would obtain language-based commands from humans (e.g. "fill the cup with water"). The other example of language-enabled robots are robots executing instruction sheets based on natural language [4], [5]. The algorithm presented in this paper, as discussed above, is developed having such robotic systems in mind as well.

In summary, we believe that this is a promising approach for automated and unsupervised generation of task-relevant training-sets for object classification/recognition, which has potential for use in many different kinds of robotic applications.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Programme and Theme: ICT-2011.2.1, Cognitive Systems and Robotics) under grant agreement no. 600578, ACAT.

REFERENCES

[1] M. Muja, R. B. Rusu, G. Bradski, and D. G. Lowe, "REIN-A fast, robust, scalable REcognition INfrastructure," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[2] M. Wail, N. Pugeault, and N. Krger, "Multi-view object recognition using view-point invariant shape relations and appearance information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[3] Y. Sun, L. Bo, and D. Fox, "Attribute based object identification," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[4] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz, "Web-enabled Robots – Robots that Use the Web as an Information Resource," *Rob. & Automat. Magazine*, vol. 18, no. 2, pp. 58–68, 2011.

[5] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth, "Robotic Roommates Making Pancakes," in *IEEE-RAS Int. Conf. on Humanoid Robots*, October, 26–28 2011, pp. 529–536.

[6] M. Tamosiunaite, I. Markelic, T. Kulvicius, and F. Wörgötter, "Generalizing objects by analyzing language," in *IEEE-RAS Int. Conf. Humanoid Robots*, oct. 2011, pp. 557–563.

[7] T. Kulvicius, I. Markelic, M. Tamosiunaite, and F. Wörgötter, "Semantic image search for robotic applications," in *Int. Workshop on Robotics in Alpe-Adria-Danube Region (RAAD2113)*, 2013.

[8] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. 264–271.

[9] ——, "A visual category filter for google images," in *Europ. Conf. Computer Vision*, May 2004, pp. 242–256.

[10] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *IEEE Int. Conf. Computer Vision*, vol. 2, oct. 2005, pp. 1816–1823.

[11] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] T. L. Berg and D. A. Forsyth, "Animals on the web," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1463–1470.

[13] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," in *Int. Conf. on Computer Vision*, Oct. 2007, pp. 1 –8.

[14] I. Khan, P. M. Roth, and H. Bischof, "Learning object detectors from weakly-labeled internet images," in *35th OAGM/AAPR Workshop*, 2011.

[15] L. Li, G. Wang, and L. Fei-fei, "Optimol: automatic online picture collection via incremental model learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[16] G. Wang, D. Hoiem, and D. Forsyth, "Building text features for object image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2009, pp. 1367 –1374.

[17] Y. Jing and S. Baluja, "Visualrank: Applying pagerank to large-scale image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1877 –1890, Nov. 2008.

[18] A. D. Holub, P. Moreels, and P. Perona, "Unsupervised clustering for google searches of celebrity images," *IEEE Int. Conf. Automatic Face and Gesture Recognition*, 2008.

[19] D. R. Dowty, L. Karttunen, and A. M. Zwicky, *Natural language parsing: Psychological, computational, and theoretical perspectives*. Cambridge University Press, 2005.

[20] M. Schoeler, S. C. Stein, A. Abramov, J. Papon, and F. Wörgötter, "Fast self-supervised on-line training for object recognition specifically for robotic applications," in *VISAPP*, 2014.

[21] Kuka Robot Systems. [Online]. Available: http://www.kuka-robotics.com

[22] M. J. Aein, E. E. Aksoy, M. Tamosiunaite, J. Papon, A. Ude, and F. Wörgötter, "Toward a library of manipulation actions based on semantic object-action relations," in *IEEE/RSJ International Conference on Intelligent Robots and System (IROS)*, 2013, p. in press.

[23] E. E. Aksoy, A. Abramov, J. Dörr, N. Kejun, B. Dellen, and Wörgötter, "Learning the semantics of object-action relations by observation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1229–1249, 2011.

[24] T. Kulvicius, K. J. Ning, M. Tamosiunaite, and F. Wörgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 145–157, 2012.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.

[26] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[27] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[28] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 2169–2178.

[29] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[30] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3360–3367.

[31] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, 1992, pp. 144–152.

[32] Y. LeCun, L. Bottou, G. Orr, and K.-R. Mller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Orr and K.-R. Mller, Eds. Springer Berlin Heidelberg, 1998, vol. 1524, pp. 9–50.

[33] S. Paschalakis, K. Iwamoto, N. Sprljan, R. Oami, and M. Bober, "The mpeg-7 video signature tools for content identification," *IEEE Trans. Circuits Syst. Video Technol.*

[34] H. Holzapfel, D. Neubig, and A. Waibel, "A dialogue approach to learning object descriptions and semantic categories," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 1004–1013, 2008.

[35] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, "Multi-view object recognition using view-point invariant shape relations and appearance information," in *International Symposium on Experimental Robotics (ISER)*, 2012.

[36] R. Deits, S. Tellex, P. Thaker, D. Simeonov, T. Kollar, and N. Roy, "Clarifying commands with information-theoretic human-robot dialog," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 58–79, 2013.