Applying Semantic Reasoning in Image Retrieval

Maaike de Boer, Laura Daniele,
Paul Brandt, Maya Sappelli
TNO
Delft, The Netherlands
email: {maaike.deboer,
laura.daniele, paul.brandt,
maya.sappelli}@tno.nl

Maaike de Boer, Maya Sappelli

Radboud University
Nijmegen, The Netherlands,
email: {m.deboer,
m.sappelli}@cs.ru.nl

Paul Brandt
Eindhoven University of
Technology (TU/e)
Eindhoven, The Netherlands
email: p.brandt@tue.nl

***Abstract*—With the growth of open sensor networks, multiple applications in different domains make use of a large amount of sensor data, resulting in an emerging need to search semantically over heterogeneous datasets. In semantic search, an important challenge consists of bridging the semantic gap between the high-level natural language query posed by the users and the low-level sensor data. In this paper, we show that state-of-the-art techniques in Semantic Modelling, Computer Vision and Human Media Interaction can be combined to apply semantic reasoning in the field of image retrieval. We propose a system, GOOSE, which is a general-purpose search engine that allows users to pose natural language queries to retrieve corresponding images. User queries are interpreted using the Stanford Parser, semantic rules and the Linked Open Data source ConceptNet. Interpreted queries are presented to the user as an intuitive and insightful graph in order to collect feedback that is used for further reasoning and system learning. A smart results ranking and retrieval algorithm allows for fast and effective retrieval of images.**

Keywords-semantics; natural language queries; semantic reasoning; image retrieval; ranking.

I. INTRODUCTION

More and more sensors connected through the Internet are becoming essential to give us support in our daily life. In such a global sensor environment, it is important to provide smart access to sensor data, enabling users to search semantically in this data in a meaningful and, at the same time, easy and intuitive manner. Towards this aim, this paper presents the GOOgle^{TM} for Sensors (GOOSE) system, which is a general-purpose search engine conceived to enable any type of user to retrieve images and videos in real-time from multiple and heterogeneous sources and sensors [1]. The proposed system especially focuses on cameras as sensors, and aims at bridging the semantic gap between natural language queries that can be posed by a user and concepts that can be actually recognized by detectors. These detectors are built using computer vision techniques, and the number of detectors is limited compared to all possible concepts that may be in the user's mind.

This work addresses the semantic interpretation of user queries to support the task of image retrieval. Our approach is general-purpose, i.e., not restricted to a specific domain, since it gives the flexibility to search for images that can contain any kind of concepts. Users can pose queries in natural language, which are parsed and interpreted in terms of objects, attributes, scenes and actions, but also semantic and spatial relations that relate different objects to each other. The system uses semantic graphs to visually explain to its users, in an intuitive manner, how a query has been parsed and semantically interpreted, and which of the query concepts have been matched to the available image detectors. For unknown concepts, the semantic graph suggests possible interpretations to the user, who can interactively provide feedback and request to train an additional concept detector. In this way, the system can learn new concepts and improve the semantic reasoning by augmenting its knowledge with concepts acknowledged by the user. Images corresponding to the recognized concepts are retrieved from video streams, ranked and presented to the user as result. The reasoning to build the semantic graphs is fully automated and uses ConceptNet [2], an external large knowledge base with concepts and semantic relations, constructed by combining multiple sources on the Web.

The main challenge in this work is the integration of several research areas in which semantics is intended in different ways. In Computer Vision, applying semantics is the process of converting elementary visual entities, e.g., pixels, to symbolic forms of knowledge, such as textual tags and predicates. In Human Media Interaction, semantics is mainly used in terms of tags used to annotate images by users. In Semantic Modelling, semantics is intended in terms of semantic models used to describe domain knowledge, such as ontologies, and inference using rules.

The goal of this paper is to show how state-of-the-art techniques in these three research areas can be combined in one single application able to semantically reason and learn, allowing its users to pose natural language queries about any topic and interact with the system to retrieve images corresponding to their queries. An overview paper about the whole GOOSE application is given in [3], where this paper is only focused on the semantic interpretation. The working of the image classification and quick image concept learning is given in [4] and fast re-ranking of visual search results is presented in [5].

This paper is structured as follows: Section II describes related work, Section III presents a short overview of the application, Section IV explains the semantic reasoning in the application, Section V contains the discussion and Section VI consists of the conclusion and future work.

II. RELATED WORK

Most of the effort in applying semantics in Computer Vision is aimed at training detectors and classifiers using large sources of visual knowledge, such as ImageNet [6] and Visipedia [7]. ImageNet is based on the WordNet [8] hierarchy of nouns, which allows to reason about *objects* in the images, but not about *actions*. Moreover, only a part of the ImageNet images is manually annotated with bounding boxes, which limits the results of the classifiers and detectors training process. Visipedia is an augmented version of Wikipedia with annotated images. Annotation is a time consuming and error-prone activity that is usually delegated to motivated crowds, who need to be trained to reduce the subjective noise in the process of image labelling. Concerning annotation, considerable effort has been spent in Human Media Interaction in labelling images for the purpose of retrieving video events. Usually, domain-specific ontologies are used as basis for annotation, such as the ontologies in [9] [10] that are used to annotate soccer games. Another example of domain-specific ontology is presented in [11] for the purpose of action recognition in a video surveillance scenario. In general, the efforts mentioned above focus on the specific algorithms for image processing and/or on the annotation of images, rather than on the semantic interpretation that should facilitate users in understanding the reasoning behind the system. Therefore, more attention should be given at integrating computer vision and semantic reasoning techniques with human interaction aspects. In this paper, three systems that integrate these aspects are discussed [12] [13] [14].

The first of these systems facilitates natural language querying of video archive databases [12]. The underlying video data model allows identification of regions (bounding boxes), spatial relations between two bounding boxes, temporal relations in terms of intervals, and trajectories. Queries are processed in terms of *objects*, *attributes*, *activities* and *events* using information extraction techniques. This is especially relevant to structure the initial user query in semantic categories that facilitate the matching with available video detectors. The query processing is realized using a link parser [15] based on a light-parsing algorithm that builds relations between pairs of words, rather than constructing constituents in a tree-like hierarchy. This is sufficient for the specific kind of word groups considered in the system [12], but is limitative for more complex queries. In contrast, a typed dependencies parser, such as the Stanford Parser [16], facilitates the processing of complex queries and allows sentences to be mapped onto a directed graph representation. In this representation, the nodes represent words in the sentence and the edges represent the grammatical relations. Moreover, the query expansion in this system [12] could benefit from a semantically richer knowledge base than WordNet [8], such as ConceptNet [2] , which is a large knowledge base constructed by combining multiple web sources, such as DBpedia [17], Wiktionary [18] and WordNet [8].

The Never Ending Image Learner (NEIL) proposed in [13] is a massive visual knowledge base that runs 24 hour a day to extract semantic content from images on the Web in terms of *objects*, *scenes*, *attributes* and their *relations*. The longer NEIL runs, the more relations between concepts detected in the images it learns. NEIL is a general-purpose system and is based on learning new concepts and relations that are then used to augment the knowledge of the system. In this way, it continuously builds better detectors and consequently improves the semantic understanding of the images. NEIL aims at developing visual structured knowledge fully automatically without human effort. However, especially in semantic reasoning, lots of knowledge stays implicit in the user's mind. Therefore, it is desirable to provide the user with mechanisms to generate feedback to improve the semantic understanding of the system. Besides the lack of a user interface for collecting feedback, NEIL does not detect *action*s. Moreover, although NEIL considers an interesting set of semantic relations, such as taxonomy (*IsA*), partonomy (*Wheel is part of Car)*, attribute associations (*Round_shape is attribute of Apple* and *Sheep is White)*, and location relations (*Bus is found in Bus_depot*), most of the relations learned so far are of the basic type *IsA* or *LooksSimilarTo*.

The work in [14] presents a video search system for retrieving videos using complex natural language queries. This system uses the Stanford Parser [16] to process the user sentences in terms of *entities, actions*, *cardinalities, colors* and *action modifiers*, which are captured into a semantic graph that is then matched to available visual concepts. Spatial and semantic relations between concepts are also considered. The system [14] is not general-purpose, but tailored for a use case of autonomous driving, which provides sufficient complexity and challenges for the video detection. This includes dynamic scenes and several types of objects. This use case limits the semantic search capability to the set of concepts that are relevant, resulting in five entity classes, i.e., *cars*, *vans*, *trucks*, *pedestrians* and *cyclists*, and fifteen action classes, such as *move*, *turn*, *park* and *walk*. The semantic graph provides an intuitive and insightful way to present the underlying reasoning to the users.

III. APPLICATION

Our application is a general-purpose search engine that allows users to pose natural language queries in order to retrieve corresponding images. In this paper, we show two visual environments in which the application has been used. As a first environment, a camera is pointed at a table top on which toy sized objects can be placed to resemble real objects. Images of (combinations of) these objects can be manually taken and sent in real time to a database. In this environment, 42 concepts and 11 attributes, which are colors, are trained using sample images in the same environment. This number can grow, because of the ability to learn new concepts.

As a second environment, we tap into highway cameras. From these cameras, images are taken continuously and are automatically processed and stored by the image classification system. At this moment, only one concept (*car*) can be detected. Up to 12 colors are available for these cars. This environment can for example be used in applications for police or defense organizations, such as following suspect cars or searching for specific accidents.

Two main use cases are supported. Firstly, users can search in historical data. Secondly, real-time images can be retrieved using notifications on outstanding queries. In the next section, we will focus on the semantic reasoning in this application.

IV. Semantic Reasoning

Figure 1 shows an overview of the system in which green and blue parts represent the components that realize the semantic reasoning, yellow parts represent the components dedicated to the image classification task and the white parts represent external components. Information about the image classification task is out of the scope of this paper, but elaborated in [4]. In the image classification, the semantics of Computer Vision is captured when the pixels are translated into annotated images.

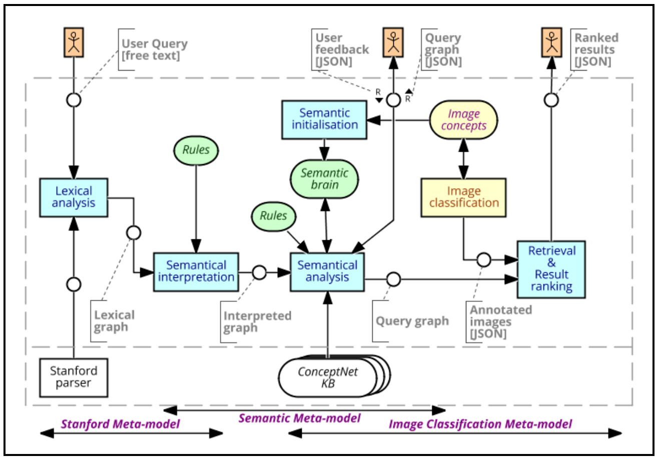


Figure 1.System overview

The input for the GOOSE system is a user query in natural language. The query is passed through four modules, while a fifth module takes care of initializing the system and learning new concepts. In the first stage, the query is sent to the Lexical Analysis module that parses it using the Stanford Parser [16], as opposed to the light link parser in [12]. The Stanford Parser returns a lexical graph, which is used as input to the Semantic Interpretation module. In this module, a set of rules is used to transform the lexical elements of the Stanford meta-model into semantic elements *objects*, *attributes*, *actions*, *scenes* and *relations*. The interpreted graph is sent to the Semantic Analysis module that matches the graph nodes against the available image concepts. If there is no exact match, the query is expanded using an external knowledge base, i.e., ConceptNet, to find a close match. The interpretation resulting from the Semantic Analysis is presented as a query graph to the user, who can interactively provide feedback used to gradually augment the Semantic Brain of the system, as inspired by NEIL [13]. The interactive part reflects the semantics of the Human Media Interaction. The query graph is inspired by the system in [14], which is, in contrast to our system, a domain-specific system. The query graph is also used as input for the Retrieval & Result ranking module, which provides the final result to the user. In the following subsections the complete process is described in detail using the example query *find an animal that is standing in front of the yellow car*.

A. Semantic Initialisation

This module provides an initial semantic capability by populating the Semantic Brain with image concepts (*objects*, *actions*, *scenes* and *attributes*) that the image classification part is capable of detecting. It also handles updates to the Semantic Brain following from new or modified image classification capabilities.

B. Lexical Analysis

In the Lexical Analysis module, the user query is lexically analyzed using the Typed Dependency parser (englishPCFG) of Stanford [16]. Before parsing the query, all tokens in the query are converted to lower case. In the example of *find an animal that is standing in front of the yellow car*, the resulting directed graph from the Lexical Analysis is shown in Figure 2.

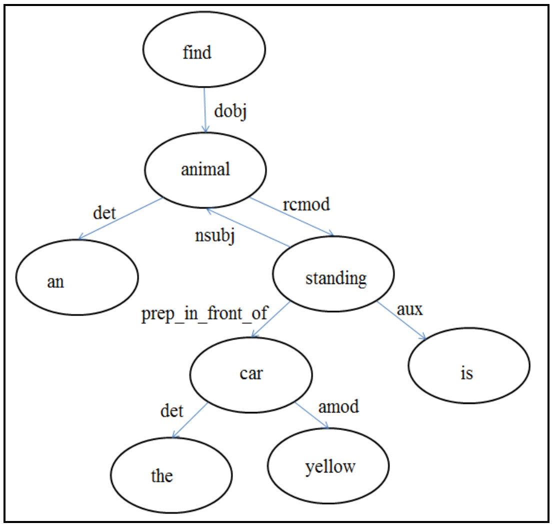


Figure 2. Lexical Graph

C. Semantic Interpretation

Since GOOSE is positioned as a generic platform, its semantics should not depend on, or be optimized for, the specifics of one single domain. Instead, we apply a generic ontological commitment by defining a semantic meta-model, shown in Figure 3, which distinguishes objects that might (i) bear attributes (*a car having a yellow color*), (ii) take part in actions (*running*), (iii) occur in a scene (*outside*), and (iv) have relations with other objects, in particular ontological relations (*a vehicle subsumes a car*), spatial relations (*an animal in front of a bus*), and temporal relations (*a bus halts after driving*). This meta-model is inspired by [12] [13] [14].

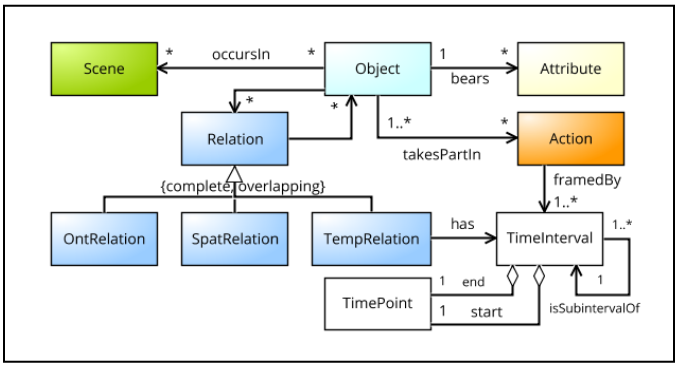


Figure 3. Semantic Meta-model

In the Semantic Interpretation module, a set of rules is used to transform the elements from the lexical graph into *objects*, *attributes*, *actions*, *scenes* and *relations*, according to the semantic meta-model in Figure 3. These rules include the following examples:

- Derive *cardinality* from a *determiner* (*det* in Figure 2), e.g., *the* in a noun in the singular form indicates a cardinality of 1, while *a/an* indicates at least 1;
- Derive *attributes* from *adjectival modifiers* (*amod* in Figure 2), i.e., adjectival phrases that modify the meaning of a noun;
- Derive *actions* from *nominal subjects* and *direct objects* (*nsubj* and *dobj* in Figure 2), i.e., the subject and object of a verb, respectively;
- Actions that represent the query command, such as *find*, *is*, *show* and *have*, are replaced on top of the tree by the subject of the sentence.

The output of the Semantic Interpretation for *find an animal that is standing in front of the yellow car* is shown in Figure 4. This is the basis of the query graph.

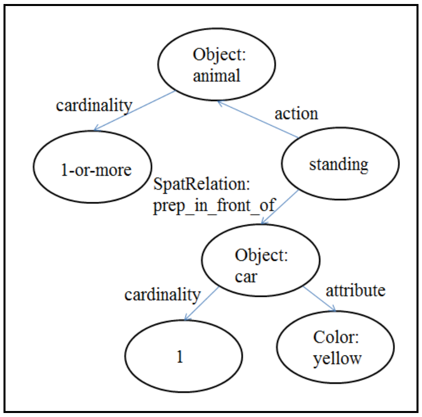


Figure 4. Interpreted Graph

D. *Semantic Analysis*

In the Semantic Analysis module, the elements from the interpreted graph, which are the query concepts, need to be matched against the concepts that can be detected by the image analysis component. The concepts that can be detected are represented by a label and stored in the system as image concepts. During the semantic analysis, the query concepts are matched against the image concepts in the Semantic Brain. If none of the objects or attributes can be detected by the image analysis module, the query concepts are expanded using ConceptNet. ConceptNet is used as opposed to WordNet in [12], because it has a more extensive knowledge base. Concept expansion is performed as follows: ConceptNet 5.2 [2] is accessed using the REST API and, among all the possible relations, we select the *IsA* relations (*OntRelation* in Figure 4) for objects, scenes or attributes, and the *Causes* relations (*TempRelation* in Figure 4) for actions. If one of the expanded objects, scenes or attributes has an exact match to one of the image concepts, that concept is added to the query graph with its corresponding relation. If there is still no match, the expansion cycle is repeated a second time. In this way, if there is, for example, no corresponding image concept for *Volvo*, this can anyway be expanded to the *car* image concept. However, when a *car* image concept is not available, the query will be further expanded in the second stage to the *vehicle* image concept. At this moment, we do not expand further than 2 iterations due to its combinatorial explosion, any potential cyclic concepts and its increasing semantic inaccuracy. The expansions, notably those originating from *IsA*, can be directed into both generalizing and specializing fashion, such that expansion of *animal* results both in *cow* as well as *creature*. Therefore, in the expanded query graph as visualized for the user, the *IsA* arrow stands for *expanded to* as opposed to a direction of subsumption. An example of the Query Graph for *find an animal that is standing in front of the yellow car* is shown in Figure 5.

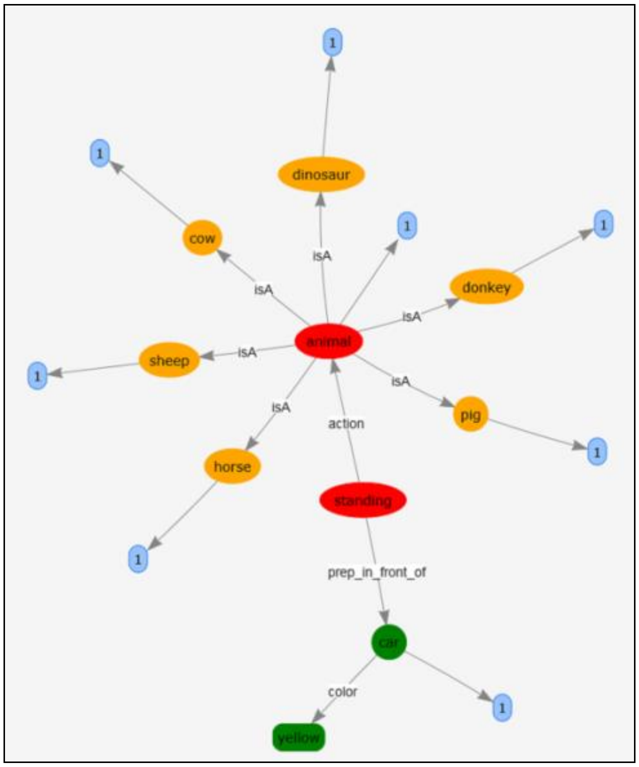


Figure 5.Example of an Expanded Query Graph

In the visualization, green colored nodes are query concepts that have a direct match with an available image concept. Red colored nodes represent query concepts that cannot be matched against an available image concept. Orange colored nodes represent suggested interpretations of query concepts using ConceptNet. For these concepts, it is uncertain whether they convey the user's intent and, therefore, require feedback from the user. Blue colored nodes represent the cardinality of the concept, e.g., the number of instances of this concept that is requested in the query. Relations between the concepts are depicted using labeled connections between the nodes corresponding to the concepts.

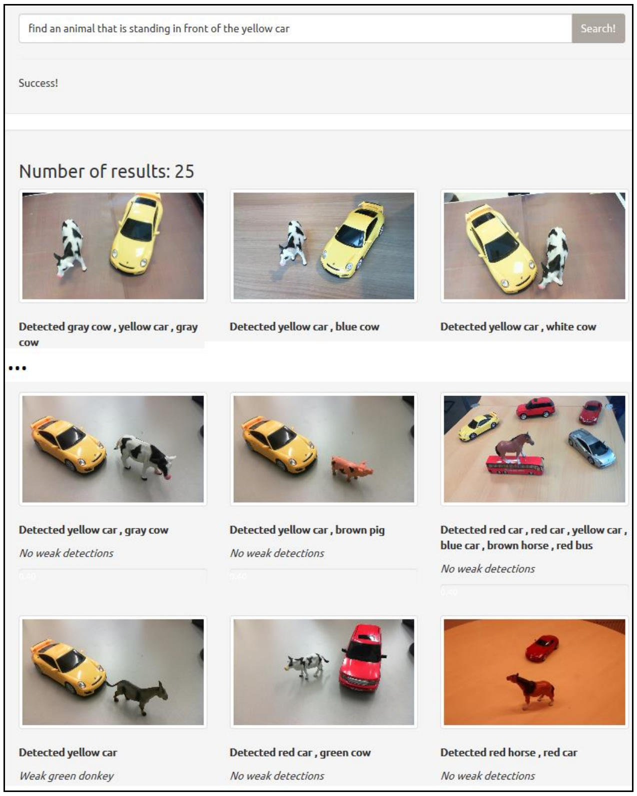
E. Retrieval and Ranking

The retrieval and ranking function need to be able to take into account the interpreted cardinality and attributes of the concepts in the query. This module retrieves those images that contain one or more concepts required by the query graph and excludes those that contain concepts that are explicitly not required (*a bicycle and no car*). The retrieval function is non-strict to ensure that there are a sufficient number of images that can be returned to the user.

The ranking on the images is based on concept, cardinality and attribute match. The motivation is that the most important elements in image search are the concepts that need to be found. For example, if a person searches for *a red car*, then it is more important that the *car* is visible in the image, and to a lesser extent whether this car is indeed *red*. Of course, this is also dependent on the context of the application.

The ranking function is penalty based. The image is included in the results if all requested concepts are present. The inverse of the confidence of the classifier for the concept, which is a value between zero and one, is taken as the basic penalty. This means that a high confidence gives a low penalty. For each concept requested in the query, a penalty of 0.5 is added to the basic penalty if 1) an attribute of a concept in the image does not match the attribute requested in the query and 2) when the image contains too few instances of the requested concept. The image is, however, not penalized if it contains too many instances. This is a choice that is also dependent on the application area. The lowest penalized images are displayed first in the results list.

Figure 6 shows the ranked result list for the query *find an animal that is standing in front of the yellow car*. In the results, we see that all images that contain a yellow car are ranked higher than those images that contain a car, but of which the attribute color is wrong, i.e., red. Even images with multiple cars of which one car is yellow are ranked higher than the images with a single car that is red. This is coherent with the interpretation of *a yellow car*, since the query states nothing explicitly about the exclusion of cars of different colors. Images with multiple cars and an animal are ranked lower than most of the images that contain a single yellow car and an animal, because the classifier is less confident that it has indeed observed a yellow car among the five vehicles in the picture.



find an animal that is standing in front of the yellow car

Success!

Number of results: 25

Detected gray cow , yellow car , gray cow

Detected yellow car , blue cow

Detected yellow car , white cow

...

Detected yellow car , gray cow

No weak detections

Detected yellow car , brown pig

No weak detections

Detected red car , red car , yellow car , blue car , brown horse , red bus

No weak detections

Detected yellow car

Weak green donkey

Detected red car , green cow

No weak detections

Detected red horse , red car

No weak detections

Figure 6. Result for Example Query

V. Discussion

During the implementation of the system we encountered various obstacles. When using semantic analysis as a method to understand which components need to be present in an image, it is important to keep in mind the limitations of the image classification.

The use of spatial relations in image search is a challenge. In terms of image analysis, objects can be positioned relative to each other using bounding boxes. An object can be left of, right of, above, under or overlapping. But, how should we interpret spatial relations such as *in front of* in the user query? With the query *animal in front of the yellow car*, is the interpretation of *in front of* based on the perspective of someone in the car, or based on the perspective of someone looking at the picture? In the former case, this would mean that an image with an animal that is to the side of the front of the car needs to be ranked higher, while in the latter case an animal that is closer to the observer (whether it is at the front, the back or the side) needs to be ranked higher. Depending on which interpretation the user wishes, the image classifier may have a higher burden, because it would need to analyze the orientation of the car, and detect on which side the front of the car is.

An additional complication concerns prepositions, such as *with*, that have ambiguous meaning. For example, the query *the woman with the red dress* is most likely interpreted as *the woman wearing a red dress*. From an image detection point of view this can be seen as *a woman who is for a large part red*. On the other hand, in the case of *the woman with the* *dog*, the interpretation of the two concepts cannot be merged. One possible solution would be to take the type of object into account (*a dress is clothing*).

Query expansion can also be a complicating factor. ConceptNet sometimes does not provide the expected relations. For example, no *IsA* relation between *animal* and *cow* exists (but a *RelatedTo*). On the other hand, a relation between *Mercedes* and *person* and *animal* is available, which should be filtered if one is looking for a car. The specific dataset that is used plays a role here. Manual additions that are specific to the dataset under consideration can be meaningful to ensure that all relevant concepts can be found during query expansion.

The combination of attributes is another point of discussion. Again, this is difficult both from the point of view of the user as well as the capabilities of the image classification. For example, the query *blue and red car* can mean that someone is searching for an image with a blue car and a red car, or that one is searching for an image with a car that is partly blue and partly red. In order to provide the required results, these kinds of ambiguities can be detected and resolved before the image classification by requesting the user to identify the correct interpretations out of the possible ones. The image classifier that we used was capable of attributing only one color to each concept, making the second interpretation impossible to detect in images.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a prototype of the GOOSE system is presented. We have shown that state-of-the-art techniques from Computer Vision, Human Media Interaction and Semantic Modelling can be combined and used in an application, while at the same time pinpointing several challenges. In the semantic part of the system, the user query is transformed into a query graph through semantic reconciliation using the Stanford Parser and ConceptNet, their meta-models and a semantic meta-model. This query graph is presented to the user in an intuitive and insightful way in order to collect feedback that is used for further reasoning and system learning.

In the future, the user should be able to have user-specific entries in the Semantic Brain. Good and bad query expansions and results are subjective and, therefore, need user-specific care.

An additional point to be further investigated concerns the semantic interpretation of the image classifiers. Here, potentially ambiguous names were used to identify these classifiers. This is in particular an issue when dealing with user-generated classifiers.

Finally, an evaluation of the semantic interpretation and result ranking modules of the GOOSE system should be performed in the future in order to validate our approach and show that our implementation can handle simple and complex queries in different domains.

ACKNOWLEDGMENT

This research was performed in the context of the GOOSE project, which is jointly funded by the enabling technology program Adaptive Multi Sensor Networks (AMSN) and the MIST research program of the Dutch Ministry of Defense.

REFERENCES

[1] R. Speer and C. Havasi “Representing general relational knowledge in conceptnet 5” LREC, 2012, pp. 3679-3686.

[2] K. Schutte et al. “GOOSE: semantic search on internet connected sensors”, Proc. SPIE, vol. 8758, 2013, pp. 875806.

[3] K. Schutte et al. “Interactive detection of incrementally learned concepts in images with ranking and semantic query interpretation”, unpublished.

[4] H. Bouma, P. Eendebak, K. Schutte, G. Azzopardi, G. Burghouts “Incremental concept learning with few training examples and hierarchical classification”, unpublished.

[5] J. Schavemaker, M. Spitters, G. Koot, M. de Boer “Fast re-ranking of visual search results by example selection”, unpublished.

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei “Imagenet: a large-scale hierarchical image database,” in IEEE CVPR, 2009, pp. 248–255.

[7] P. Perona “Visions of a Visipedia”, Proc. of IEEE, 98.8, 2010, pp. 1526-1534.

[8] C. Fellbaum “WordNet”, Blackwell Publishing Ltd, 1998.

[9] L. Bai1, S. Lao, G.J.F. Jones, and A.F. Smeaton “Video Semantic Content Analysis based on Ontology”. In Int. Machine Vision and Image Processing Conf., 2007, pp. 117-124.

[10] A.D. Bagdanov, M. Bertini, A. Del Bimbo, G. Serra, C. Torniai “Semantic annotation and retrieval of video events using multimedia ontologies”. In Int. Conf. on Semantic Computing, 2007, pp. 713-720.

[11] A. Oltramari and C. Lebiere “Using Ontologies in a Cognitive-Grounded System: Automatic Recognition in video Surveillance”, In Proc. 7 Int. Conf. on Semantic Technology for Intelligence, Defense, and Security, 2012.

[12] G. Erozel, N. K. Cicekli, I. Cicekli “Natural language querying for video databases,” Information Sciences, 178.12, 2008, pp. 2534–2552.

[13] X. Chen, A. Shrivastava, and A. Gupta “NEIL: Extracting Visual Knowledge from Web Data”, IEEE Int. Conf. on Computer Vision, 2013, pp. 1409-1416.

[14] D. Lin, S. Fiedler, C. Kong, R. Urtasun “Visual Semantic Search: Retrieving Videos via Complex Textual Queries”. In IEEE Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2657-2664.

[15] D. Sleator and D. Temperley, “Parsing English with a Link Grammar”, In Third Int. Workshop on Parsing Technologies, 1993.

[16] M.-C. de Marneffe, B. MacCartney, C. D. Manning. “Generating Typed Dependency Parses from Phrase Structure Parses”, LREC, 2006, pp. 449-454.

[17] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives “Dbpedia: A nucleus for a web of open data” Springer Berlin Heidelberg, 2007, pp. 722-735.

[18] E. Navarro et al, “Wiktionary and NLP: Improving synonymy networks”. In Proc. of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, 2009, pp. 19-27, Association for Computational Linguistics.

ISBN: 978-1-61208-445-9