

Image-Based World-perceiving Knowledge Graph (WpKG) with Imprecision

Navid Rezaei¹, Marek Z. Reformat^{$1,2(\boxtimes)$}, and Ronald R. Yager³

 ¹ University of Alberta, Edmonton, T6G 1H9, Canada {nrezaeis,marek.reformat}@ualberta.ca
² University of Social Sciences, 90-113 Lódź, Poland
³ Iona College, New Rochelle, NY 10801, USA yager@panix.com

Abstract. Knowledge graphs are a data format that enables the representation of semantics. Most of the available graphs focus on the representation of facts, their features, and relations between them. However, from the point of view of possible applications of semantically rich data formats in intelligent, real-world scenarios, there is a need for knowledge graphs that describe contextual information regarding realistic and casual relations between items in the real world.

In this paper, we present a methodology of generating knowledge graphs addressing such a need. We call them *World-perceiving Knowledge Graphs* – WpKG. The process of their construction is based on analyzing images. We apply deep learning image processing methods to extract scene graphs. We combine these graphs, and process the obtained graph to determine importance of relations between items detected on the images. The generated WpKG is used as a basis for constructing possibility graphs. We illustrate the process and show some snippets of the generated knowledge and possibility graphs.

Keywords: Knowledge graph \cdot Deep learning \cdot Common sense \cdot Possibility theory

1 Introduction

Knowledge graphs are composed of a set of triple relations, i.e. $\langle subject - predicate - object \rangle$, where subjects and objects are items connected via predicates representing relations between them. The graphs are useful in representing data semantics and are employed in different applications, such as common-sense and causal reasoning [1,2], question-answering [3], natural language processing [4], and recommender systems [5]. Some examples of existing knowledge graphs are DBpedia [6], Wikidata [7], Yago [8], the now-retired Freebase [9], and WordNet [10]. The aforementioned knowledge graphs contain information about facts,

© Springer Nature Switzerland AG 2020

M.-J. Lesot et al. (Eds.): IPMU 2020, CCIS 1237, pp. 415–428, 2020. https://doi.org/10.1007/978-3-030-50146-4_31 their features, and basic relations between them. They focus on people, geographical locations, movies, music, and organizations and institutions. They are missing a piece of information about everyday real-world items, their contexts, and arrangements.

From the human perspective, we can state that the visual information plays a significant role in human learning processes [11]. At the same time, the eye's information transfer rate is quite high [12] that makes a visual stimulus to be of significant importance in processes of gaining understanding about different items and how they are related to each other. Given the importance of visual data, it is appealing to develop systems that could observe, learn and create knowledge based on such data. Additionally, traditional knowledge graphs do not provide any degree of confidence associated with relations. It is assumed that all of them are equally important.

In this paper, we look at the task of creating knowledge graphs based on visual data. The idea is to process images, generate scene graphs from them, and aggregate these graphs. Graphs constructed in such a way contain knowledge about everyday objects, their contexts and their situational information, as well as information related to the importance of common-sense relations between multiple objects in their natural scenarios.

We call such a graph *World-perceiving Knowledge Graph*, WpKG in short. The quality and suitability of knowledge we retrieve from images depend on the capability of tools and methods we use for image processing. Processing an image means generating a scene graph representing relations between objects/entities present on this image. Once numerous images are processed, all scene graphs are aggregated. This alone allows us to treat the process of constructing graphs via aggregation as the human-like process of learning via processing of observed images.

We also look at a process of using knowledge graphs – WpKGs – to construct possibility graphs reflecting conditional dependencies between sets of entities as observed in their usual environments. The information about the importance of relations allows us to build possibilistic conditional distributions. They are used for processing and reasoning about entities and relations between them in their own relevant contexts. The included case study shows an application of the presented procedure to Visual Genome (VG) dataset [13].

2 Related Work

Extracting information from different media to create a knowledge graph has been examined in the literature. Yet, the area of focus of these works has been different: some of them focus on images, some on text, and some on a combination of both. Also, the methods used for information retrieval can be different – automatic or manual. A brief overview is presented in Subsect. 2.1.

Possibilistic knowledge bases and graphs are important forms representing uncertainty of data and information [14], and [15]. A set of basic definitions is included in the following subsections.

2.1 Knowledge Graph Construction

There is a number of different knowledge graph generation methods that focus on text as the source of information, such as NELL [16], ConceptNet [17], ReVerb [18], and Quasimodo [19]. Some other published approaches, such as WebChild KB [20,21] or LEVAN [22], extract knowledge from text and image captions or only from image objects without in-image relations. Probably, the most relevant work to our work is NEIL [23], which create a knowledge graph directly from images.

Compared to NEIL, our proposed automatic approach is capable of extracting much more types of object-to-object relations. Compared to ConceptNet, which represents an example of a semi-automatic method of retrieving knowledge from text, our proposed approach can extract common-sense relations based on only observing visual data.

2.2 Possibilistic Knowledge Base

A possibilistic base is a set of pairs (p, α) where p is a proposition, and α is a degree to which p is true and is in the interval (0, 1) [14]. Let Ω be a set of interpretations of the real world, and possibilistic distribution π a mapping from Ω to the interval (0, 1). An interpretation ω that satisfies p has $\pi(\omega) = 1$, and $1 - \alpha$ when ω fails to satisfy p. In summary:

$$\forall \omega \in \Omega, \pi_{\{p \ \alpha\}}(w) = 1 \qquad if \ \omega \models p$$
$$= 1 - \alpha \ otherwise$$

From now on, we identify the base as $\sum = \{(p_i, \alpha_i), i = 1, \dots, n\}$. Then all interpretations satisfying propositions in \sum have the possibility degree of 1, while other interpretations are ranked based on the highest values of α associated with proposition they do not satisfy, i.e., $\forall \omega \in \Omega$:

$$\pi_{\sum}(w) = 1 \qquad \qquad if \ \omega \models \sum_{i=1 - \max\{\alpha_i : (p_i, \alpha_i) \in \sum \text{ and } \omega \models \neg p_i\} \text{ otherwise}}$$

In other words, π_{\sum} induces a necessity 'grading' of p_i that evaluates to what extent p_i is a consequence of the available knowledge. The necessity measure *Nec* is:

$$Nec_{\pi_{\Sigma}}(p_i) = 1 - max\{\pi_{\Sigma}(\omega) \mid \omega \models \neg p_i\}$$

Based on that, we can say that (p_i, α_i) is a plausible conclusion of π_{\sum} if

$$Nec_{\pi_{\Sigma}}(p_i) > Nec_{\pi_{\Sigma}}(\neg p_i)$$

and $Nec_{\pi_{\Sigma}}(p_i) \geq \alpha_i$ [24].

A possibility distribution π_{Σ} is *normal* if there is an interpretation ω that it totally possible, i.e., $\pi_{\Sigma}(\omega) = 1$.

2.3 Possibilistic Graph

A possibility graph ΠG is an acyclic directed graph [14]. The nodes of such a graph are associated with variables A_i , each with its domain D_i ; while its edges represent dependencies between elements of nodes. For the case of binary variables, i.e., when $D_i = \{a_i, \neg a_i\}$, the assignment of value to the variable is called an interpretation ω . Let us denote a set of nodes that have edges connecting them to a node A_i as its parents: $Par(A_i)$. Possibility degrees Π associated with nodes are:

- for each node A_i without a parent $Par(A_i) = 0$ prior possibility degrees associated with a single node are $\Pi(a)$ for every value $a \in D_i$ of the variable A_i ; possibilities must satisfy the normalization condition: $max_{a \in D_i} :$ $\Pi(a) = 1.$
- for each node A_j with parent(s) $Par(A_j) \neq 0$ possibility degrees are conditional ones $\Pi(a|\omega_{Par(A_j)})$ where $a \in D_j$, and $\omega_{Par(A_j)}$ is an element of the Cartesian product of domains D_k of variables $A_k \in Par(A_j)$; as above, conditional possibilities must satisfy the normalization condition: $max_{a \in D_j} : \Pi(a|\omega_{Par(A_j)}) = 1.$

In our case, a conditional probability measure is defined using min:

$$\Pi(p|q) = 1 \qquad if \ \Pi(q \land p) = \Pi(q)$$
$$= \Pi(q \land p) \ otherwise$$

and obeys [14]:

$$\Pi(q \wedge p) = \min\{\Pi(p|q), \Pi(q)\}$$

3 Generation of Image-Based WpKG

We introduce a systemic approach to generate knowledge graphs given visual data. Such graphs provide us with contextual information about objects present in the world with very limited input from humans. There are unique challenges associated with the generation of this type of graph. First, we need methods able to detect objects in images, and second, we require tools to extract relations between the detected objects.

Once we have the object recognition and relation extraction processes, we execute them on a set of images. The obtained triples - < entity - relation - entity> are aggregated into a single knowledge graph. The strength of relations is determined by the number of co-occurrences of objects with specific relations. The overall process is shown in Fig. 1.

Having a trained model, the process is liberated from specific visual data and its annotations. Additionally, more visual data can be processed using the proposed methodology and comprehensive context-specific knowledge graphs could be created.



Fig. 1. Overall procedure for generation of a knowledge graph from images

3.1 Detection of Objects

To detect objects and their corresponding bounding boxes, we use the Faster R-CNN model [25]. In this model, the full image is passed through a convolutional neural network (CNN) to generate image features. To detect image features, usually a pre-trained CNN, such as VGG network [26], trained on ImageNet [27] is used. Given the image features as input, another neural network, called Region Proposal Network (RPN), predicts regions that may contain an object and their corresponding bounding boxes. This learning network is the principal contribution of the Faster R-CNN model compared to the Fast R-CNN model [28]. This results in an improvement of performance in both training and inference. The regions of interest (RoIs) are then mapped into the image feature tensor, and via application of a process called RoI Pooling the regions are downsampled to be fed to the next neural network. This allows for the prediction of image classes and their correct bounding boxes. Given the error losses from the classification and bounding box predictions, the entire network is trained end-to-end using backpropagation and stochastic gradient descent (SGD) [29]. An illustration of the process can be found in Fig. 1.

3.2 Identification of Relations Between Objects

Determining relations between objects is required to generate scene graphs and it can be done in several ways. There has been several publications that propose such methods as Iterative Message Passing [30], Neural Motifs [31], Graphical Contrastive Losses [32], and Factorizable Net [33]. In our work, we use the Iterative Message Passing model.

The Iterative Message Passing model predicts relations between objects detected by the Faster R-CNN model. Mathematically, a scene graph generation process means finding the optimal $\mathbf{x}^* = \arg \max_{\mathbf{x}} \Pr(\mathbf{x}|I, B_I)$ that maximizes the following probability function:

$$\Pr(\mathbf{x}|I, B_I) = \prod_{i \in V} \prod_{j \neq i} \Pr(x_i^{cls}, x_i^{bbox}, x_{i \to j}|I, B_I).$$
(1)

where I is an image, B_I represents proposed object boxes, \mathbf{x} is a set of all variables, including classes, bounding boxes and relations ($\mathbf{x} = \{x_i^{cls}, x_i^{bbox}, x_{i \to j} | i = 1 \dots n, j = 1 \dots n, i \neq j\}$), with n representing the number of proposed boxes, x_i^{cls} as a class label of the *i*-th proposed box, x_i^{bbox} as the offset of bounding box relative to the *i*-th proposed box, and $x_{i \to j}$ as a predicate between the *i*-th and *j*-th proposed boxes.

3.3 Aggregation of Scene Graphs

The process of amalgamating generated image scene graphs that results in a single knowledge graph has a number of challenges: 1) establishing a unique identifier for each entity; 2) identifying the importance of connections; 3) dealing with missing values and incorrect data; and 4) keeping the knowledge graph updated in presence of new data.

In the specific case of the Visual Genome dataset, we use *synsets* from Word-Net to identify nodes and relations, as well as different meanings of a specific word. There are various methods to identify the uniqueness of words, such as using words occurring in natural language, grouping similar words with the same meaning, or trying to assign words to their specific synsets. Yet, another way is to keep words and phrases as they are and let their occurrence numbers show the importance of connections and nodes. Such a simple approach provides a good indication which relations are more likely to occur.

Another challenge is to mitigate missing or incorrect information. For example, the used methods/models could incorrectly label objects/relations and the processes could fail to find unique words or synsets. Even the hand-annotated data in the Visual Genome (VG) dataset [13], which is used for training, has missing and incorrect data [34]. The unknowns are reduced by relying on the information already present, such as recovering a missing synset based on an already-known name to synset relation or WordNet.

4 Image-Based WpKG: Experimental Studies

The Iterative Message Passing model [30] is trained on the VG dataset. It contains 108,077 images that capture everyday scenarios. For evaluation, only the most common 150 object categories and 50 predicates are used.

The Faster R-CNN model that is applied to detect objects and their bounding boxes is pre-trained on MS-COCO dataset. This dataset has 80 object categories. The training set is of size 80k images. Validation and test sets are 40k and 20k images, respectively. Around thirty percent of the VG dataset (test set) is used to detect objects and predict predicates. The subset has around 30,000 images. Running the process described in Sect. 3, a WpKG with 138 nodes and 7,287 relations is generated.

Neo4j [35] software is used to store and analyze the generated graph. It allows us to store object and relationship names and synsets, as well as occurrence numbers. Also, it visualizes a structure composed of triples *subject-predicateobject*. One advantage of the generated WpKG is the existence of common sense relations occurring in the actual world extracted during the processing of visual data. The most important entities related to the entity of interest can be found by inspecting the strength of connections between them. One way to accomplish this is to measure how often these objects are associated with each other.

As an example, the entity *plate* together with the related entities is shown in Fig. 2 (a). As we can see, removing non-frequent relations leads to identification of tightly related objects relevant to the *plate*, Fig. 2 (b).



Fig. 2. Relationships to/from *plate* entity: with at least one instance and interrelationships between associated items (a); and at least 10 instances for each relationship and without the interrelationships (b).

A sample of relation occurrence statistics is shown in Table 1. Based on the analysis of visual data, we can find out about some common-sense knowledge, such as places where a *vase* can be placed, and what can be put into it. Most of the relations, such as flower-in-vase, make sense and agree with the crowd-sourced VG dataset. However, some relations, such as vase-in-vase, may not make sense. This could be a shortcoming of the method/model used for prediction of relations. Besides a better model, processing more images and detecting more types of relations and objects may improve the results.

The comparison of our method, which is based on image processing, with other relevant automatic and semi-automatic methods is demonstrated in Table 2.

	Subject	Predicate	Object	Occurrence number
	Woman	Wearing	Shirt	192
Woman	Woman	Holding	Umbrella	168
	Woman	Has	Hair	141
	Plate	On	Table	388
Plate	Plate	On	Plate	193
	Plate	On	Pizza	19
	Flower	In	Vase	173
Flower	Flower	On	Table	41
	Flower	On	Tree	15
	Vase	On	Table	116
Vase	Vase	In	Vase	44
	Vase	Has	Flower	31

Table 1. Three most common relations in WpKG generated using Faster R-CNN and iterative message passing models to recognize objects and predict relations, respectively.

5 WpKG-Based Possibilistic Graph and Base

The generated WpKGs consist of an enormous amount of nodes and relations. The relations – as built via aggregation of scene graph relations – contain information about the frequency of occurrence. This means that each relation is equipped with a weight indicating its strength and importance. For practical use, WpKG can be further processed and a subset of nodes together with relations between them can be used to construct a possibilistic graph.

Table 2. Comparison of relevant generated knowledge graphs from literature. Ourmethod and NEIL are the ones that focus on in-image relations.

Method	In-image	Input source(s)	out source(s) Relation types		Automation
	relations				
NEIL [23]	Yes	Image	<10	$<\!10K$	Automated
ConceptNet [17]	No	Text	<100	34M	Semi-automated
LEVAN [22]	No	Text and	<10	< 100 K	Automated
		Objects in			
		Images			
WebChild KB 2.0 [21]	No	Text and	>1000	>18M	Automated
		Image/Video			
		Captions			
Quasimodo [19]	No	Text (logs and	Dynamic	2.3M	Automated
		QA forums)			
Our work	Yes	Image	<50 (Dynamic)	>7K	Automated

5.1 Extracting Possibilistic Graph from WpKG

A WpKG is constructed with no constrains. It contains cycles, very strong and weak relations, as well as erroneous information due to the imperfection of used image processing tools. In that context, a possibilistic graph is more organized and 'clean'. Therefore, extracting nodes and edges from WpKG and building a graph that satisfies rules of the possibilistic graph (Sect. 2.3) seems important steps in utilizing generated WpKGs.

First, a proto-possibilistic graph is constructed. It is free of cycles and contains outwards relations linked to the entity of interest. The procedure used to extract relevant entities and connections is presented as Algorithms 1 and 2. The important aspects of this process are:

- Algorithm 1, line 4 the value of *Depth* identifies the allowed length of a 'relation chain' at the process of building a graph;
- Algorithm 2, line 6 the procedure *randomize_createGroups()* is crucial in the construction process: 1) randomization of a sequence of entities allows to generate graphs with different paths; once this is combined with a process presented in line 8 (explained below) it prevents the existence of cycles in the generated graph; 2) grouping of relations/predicates connected to the same object, i.e., prepositions/adjectives playing the role of relations; as illustration, see entities *flower, window, table, plant*, Fig. 3;
- Algorithm 2, line 8 this allows to solve an issue of cycles, i.e., relations between pairs of entities *flower-vase*, *plant-vase* and *table-vase*, Fig. 3, would lead to cyclic directed graph; however, if a connection between both entities already exist, a new one in the opposite direction is not created.

Alg	Algorithm 1: Construction of Proto-Possibilistic Graph					
D)ata:	Image-based WpKG ; Seed_Entity; Depth				
R	lesult:	Proto-Possibilistic Graph				
1 b	\mathbf{egin}					
2	$root \star$	$-Seed_Entity;$				
3	$d \leftarrow 1$;				
4	call C	TreateConn(root, d, Depth);				
5	_ retur	n;				

The application of the presented procedure leads to a graph that is acyclic and direct. It also contains occurrences associated with each connection. The last step of constructing a possibilistic graph is to determine possibility degrees. To do so, all input connections to a given node are analyzed. The maximum value is identified and is used for normalization of all other occurrence values associated with inward connections to the node. This ensures satisfaction of the requirement of maximum possibility equal to 1.0 (Sect. 2.3).

Algorithm 2: Generation of Connections

1	CreateConn(r, d, Depth)
2	begin
3	if $d \leq Depth$ then
4	$listChild \leftarrow create_list_children(r);$
5	if $listChild \neq \emptyset$ then
6	$randomize_createGroups(listChild);$
7	for $e \in listChild$ do
8	if e not_connected_to r then
9	setupConnection(r, e);
10	$ \begin{tabular}{ c c c } \hline call \ CreateConn(e,d+1,Depth); \end{tabular} \end{tabular} \end{tabular}$
11	else
12	
13	else
14	return ;
15	return;

5.2 Construction of Possibilistic Base

The extracted possibilistic graph allows us to build a possibilistic knowledge base. Here, we follow the process presented in [14]. For that purpose, we consider the graph as a set of triples:

$$\Pi G = \{(a, P_a, \alpha) : \Pi(a | P_a) = \alpha\}$$

where a is an instance of A_i and P_a is the Cartesian product of domains D_k of variables $A_k \in Par(A_i)$. Each such triple can be represented as a formula:

$$(\neg a \lor \neg P_a, 1 - \alpha)$$

so, following [14], we have that the possibilistic knowledge base associated with ΠG defined as:

$$\sum = \{ (\neg a_i \lor \neg P_{a_i} 1 - \alpha_i) / (a_i, P_{a_i}, \alpha_i) \in \Pi G \}$$

6 Possibilistic Graph and Base: Experimental Studies

Let us illustrate the process of building a simple possibilistic graph and a possibilistic knowledge base. We apply the procedure to build a graph of facts related to the entity *vase*, and relations between this entity and other entities from the vase's environment.

Application of Algorithm 1 to the generated WpKG allows us to extract entities related to the entity of interest, vase. The Neo4j snapshot of WpKG with vase and relations to 'relevant' entities is shown in Fig. 3(a). The version processed by the algorithm is shown in Fig. 3(b). It contains – marked as dashed lines – the pairs flower-vase, plant-vase, and table-vase that could result in different graphs depending on the element of randomness embedded in the procedure randomize_createGroups(), Algorithm 2.

The WpKG with occurrences assigned to connections allows us to determine conditional degrees. We have simplified our graph, i.e, combined all inward connections to a node into a single one, as shown in Fig. 3(c). This graph is further processed – the occurrence numbers are used to determine possibility values. Based on the graph in Fig. 3(c), we build conditional possibility degrees. All of them are presented in Tables 3, 4, and 5.



Fig. 3. A fragment of WpKG and a possibilistic graph constructed based on it.

$(a) \Pi(Vaga)$			(b) l	∏(Counter	Vase	e)	
(a) II(vase)			Cou	nter Vase	vase	-	vase
vase 1.			C	ounter	1.		1.
\neg vase 1.			\neg counter				1.
(c) Π (Flower	(Vase))		(d) Π(Pla	V	ase)
Flower Vase	vase	\neg vase		Plant Vas	se va	se	\neg vas
flower	1.	1.		plant	1		1.
\neg flower	1.	1.		\neg plant	1		1.

Table	3. F	Possibility	degrees	for	Vase,	Flower,	Counter,	and	Plant.
-------	------	-------------	---------	-----	-------	---------	----------	-----	--------

Window Vase, Flower	$\mathbf{vase},\neg\mathrm{flower}$	$\neg vase, flower$	Elsewhere	
window	1 .545		1.	
¬window	1.	1.	1.	

Table 4. Possibility degrees for $Window - \Pi(Window | Vase, Flower)$

Table 5. Possibility degrees for *Table* $-\Pi$ (Table|Window, Vase, Flower, Counter)

Shelf Window,	window,	¬window,	¬window,	¬window,	Elsewhere
Vase, Flower,	\neg vase,	vase,	\neg vase,	¬vase,	
Counter	\neg flower,	¬flower,	flower,	\neg flower,	
	\neg counter	\neg counter	\neg counter	counter	
table	.017	1.	.347	.017	1.
¬table	1.	1.	1.	1.	1.

The last step of our case study is dedicated to the construction of a possibilistic knowledge base, Sect. 5.2. As a result, we obtain:

 $\sum = \{ (\neg window \lor vase \lor \neg flower, .455), \\ (\neg table \lor \neg window \lor vase \lor flower \lor counter, .983), \\ (\neg table \lor window \lor \neg vase \lor flower \lor counter, .653), \\ (\neg table \lor window \lor vase \lor flower \lor \neg counter, .983) \}.$

7 Conclusion

The paper focuses on the automatic construction of a knowledge graph – called *World-perceiving Knowledge Graph (WpKG)* – that contains results of the analysis of multiple images. Further, the generated WpKG is processed and multiple possibilistic graphs can be constructed based on it.

It is shown that using deep learning models, we can extract common-sense situational information about objects present in visual data. The trained neural networks may already know these relations implicitly, but extracting this knowledge in the form of a knowledge graph provides the ability to have this information explainable and explicit. The strength of the overall procedure depends on the capabilities of the applied learning model as well as the data it has been trained on. By improving the models themselves, the overall procedure can be improved.

Constructed WpKGs are contextualized by images used as an input to the presented process. A different graph will be obtained when images representing a specific geographical location are used, while a different graph will be built based on images illustrated a specific historical event. Also, multiple different possibilistic graphs can be created to reason about the correctness of contextual utilization of specific items and relations between them.

Given the adaptability of WpKG to new scenarios, context-aware and even time-variant knowledge graphs can be constructed. For example, processing car images from a specific country will lead to the construction of WpKG representing a very specific information related to cars' details and their contextual settings. Another important aspect that can be considered is time. It can affect both occurrences of relations and meanings of words linked to the nodes.

As future work, better models can be used to improve the overall construction process, biases can be reduced by implementing procedures to diversify the input images, and prediction of unknown objects can be added.

References

- Sap, M., et al.: ATOMIC: an atlas of machine commonsense for if-then reasoning. In: AAAI, pp. 3027–3035 (2019)
- Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: commonsense transformers for automatic knowledge graph construction. In: 57th ACL Meeting, pp. 4762–4779 (2019)
- Hao, Y., et al.: An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In: 55th ACL Meeting, pp. 221– 231 (2017)
- 4. Kapanipathi, P., et al.: Infusing knowledge into the textual entailment task using graph convolutional networks. In: AAAI (2020)
- Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B., Lee, D.L.: Billion-scale commodity embedding for e-commerce recommendation in Alibaba. In: KDD 2018 (2018)
- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: The Semantic Web, pp. 722–735 (2007)
- Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM 57(10), 78–85 (2014)
- Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., Weikum, G.: YAGO: a multilingual knowledge base from Wikipedia, Wordnet, and Geonames. In: Groth, P., et al. (eds.) International Semantic Web Conference, pp. 177–185 (2016)
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: ACM SIG-MOD, pp. 1247–1250 (2008)
- Miller, G.A.: WordNet: a lexical database for English. Commun. ACM 38, 39–41 (1995)
- 11. The importance of vision. Ophthalmology **94**, 9–13 (1987). https://doi.org/10. 1016/S0161-6420(87)33553-5
- Jacobson, H.: The informational capacity of the human eye. Science 113(2933), 292–293 (1951)
- Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. 123, 32–73 (2016)
- Benferhat, S., Dubois, D., Garcia, L., Prade, H.: Possibilistic logic bases and possibilistic graphs. In: 15th Conference on Uncertainty in AI, pp. 57–64 (1999)
- Gebhardt, J., Kruse, R.: Automated construction of possibilistic networks from data. J. Appl. Math. Comput. Sci. 6(3), 101–136 (1996)

- 16. Mitchell, T., et al.: Never-ending learning. Commun. ACM 61, 103–115 (2018)
- 17. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: an open multilingual graph of general knowledge. In: AAAI, pp. 4444–4451 (2017)
- Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Conference on Empirical Methods in NLP, pp. 1535–1545 (2011)
- Romero, J., Razniewski, S., Pal, K., Pan, J.Z., Sakhadeo, A., Weikum, G.: Commonsense properties from query logs and QA forums. In: CIKM 2019, pp. 1411– 1420 (2019)
- Tandon, N., de Melo, G., Suchanek, F.M., Weikum, G.: WebChild: harvesting and organizing commonsense knowledge from the web. In: WSDM 2014, pp. 523–532 (2014)
- Tandon, N., de Melo, G., Weikum, G.: WebChild 2.0 : fine-grained commonsense knowledge distillation. In: 59th ACL Meeting, pp. 115–120 (2017)
- Divvala, S.K., Farhadi, A., Guestrin, C.: Learning everything about anything: Webly-supervised visual concept learning. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3270–3277 (2014)
- Chen, X., Shrivastava, A., Gupta, A.: NEIL: extracting visual knowledge from web data. In: 2013 IEEE International Conference on Computer Vision, pp. 1409–1416 (2013)
- Benferhat, S., Sossai, C.: Merging uncertain knowledge bases in a possibilistic logic framework. In: 14th Conference on Uncertainty in AI, pp. 8–15 (1998)
- Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149 (2015)
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a largescale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
- Girshick, R.: Fast R-CNN. In: IEEE Conference on Computer Vision, pp. 1440– 1448 (2015)
- LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. 1(4), 541–551 (1989)
- Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3097–3106 (2017)
- Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: scene graph parsing with global context. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5831–5840 (2018)
- Zhang, J., Shih, K.J., Elgammal, A., Tao, A., Catanzaro, B.: Graphical contrastive losses for scene graph parsing, pp. 11535–11543 (2019)
- Li, Y., Ouyang, W., Zhou, B., Shi, J., Zhang, C., Wang, X.: Factorizable net: an efficient subgraph-based framework for scene graph gen. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 335–351 (2018)
- 34. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., Ling, M.: Scene graph generation with external knowledge and image reconstruction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1969–1978 (2019)
- 35. Neo4j Inc: Neo4j. https://neo4j.com