

# Multimodal Question Answering over Structured Data with Ambiguous Entities

Huadong Li\*  
Shandong University  
Jinan  
Shandong, China

Yafang Wang\*†  
Shandong University  
Jinan  
Shandong, China

Gerard de Melo  
Rutgers University  
New Brunswick  
USA

Changhe Tu  
Shandong University  
Jinan  
Shandong, China

Baoquan Chen  
Shandong University  
Jinan  
Shandong, China

## ABSTRACT

In recent years, we have witnessed profound changes in the way people satisfy their information needs. For instance, with the ubiquitous 24/7 availability of mobile devices, the number of search engine queries on mobile devices has reportedly overtaken that of queries on regular personal computers. In this paper, we consider the task of multimodal question answering over structured data, in which a user supplies not just a natural language query but also an image. Our system addresses this by optimizing a non-convex objective function capturing multimodal constraints. Our experiments show that this enables it to answer even very challenging ambiguous entity queries with high accuracy.

## Keywords

Question Answering; Multimodal; Multimedia Knowledge Bases

## 1. INTRODUCTION

**Motivation.** The way people seek information has evolved substantially in recent years. With the ubiquitous 24/7 availability of Internet-connected mobile devices, including smartphones, tablet computers, and augmented reality glasses, the way we rely on computing in our daily lives has undergone changes in numerous important respects. With regard to information retrieval, the number of search engine queries on mobile devices has reportedly surpassed the corresponding number for regular personal computers.

One important ramification is that mobile device usage tends to favour other input modalities than the traditional keyboard and mouse interface. Touch interfaces can directly substitute for some of the previous forms of interaction. Yet, typing on mobile devices can be cumbersome, especially on the go, as evidenced by the

\* The first two authors contributed equally to the paper.

† Contact: yafang.wang@sdu.edu.cn

©2017 International World Wide Web Conference Committee (IW3C2), published under Creative Commons CC BY 4.0 License.  
WWW 2017, April 3–7, 2017, Perth, Australia.  
ACM 978-1-4503-4914-7/17/04  
<http://dx.doi.org/10.1145/3041021.3054173>



phenomenon of very short emails with *Please excuse brevity*-style disclaimers.

At the same time, these new device categories open up significant new opportunities for more natural forms of interaction. For one, due to advances in speech recognition, mobile device users have been issuing increasingly longer queries. Indeed, it is now becoming commonplace for queries to be full-fledged questions. Additionally, smartphones, augmented reality glasses, and other devices are equipped with cameras enabling us to digitally capture whatever we are visually perceiving in the form of an image. Alternatively, especially for items we are not directly perceiving, stylus and touch input also enables us to sketch things that we may be looking for, or that are relevant to the query at hand, enabling a novel but little-explored information seeking paradigm.

Indeed, humans sometimes encounter difficulties in formulating a natural language query specific enough to make the desired answer evident and sufficiently unambiguous. Considering the popular notion of *a picture being worth a thousand words*, novel input modalities for search may thus in fact be more than just an alternative. We conjecture that in certain cases, additional multimodal input may substantially aid the user in establishing and conveying their search intent to the question answering engine. This applies to visual perception of objects or circumstances that may be hard to express verbally but easy to capture as a picture. This likewise also applies to mental imagery, which we may have trouble conveying verbally, while still being able to sketch it.

**Contribution.** In this paper, we consider the novel task of multimodal question answering (QA). As shown in Figure 1, users seeking information supply not just a natural language query but also a relevant photo or sketch. We focus on questions and answers that can be addressed using structured knowledge repositories. Our system tackles this challenging problem in multiple steps. First, we apply regular linguistic analysis methods to the natural language part of the query. Our experiments show that this component of our system alone already delivers competitive results comparable to those of current question answering systems. Subsequently, we draw on a novel algorithm based on optimizing a non-convex objective function with linear constraints. This allows us to jointly capture both linguistic and multimodal constraints in a single joint optimization problem. The algorithm derives a structured query that is then used to obtain the final answers from the knowledge store.

Our experiments are conducted on a set of particularly challenging questions involving ambiguous entities. The results show that our

algorithm enables our system to succeed even on particularly difficult queries that humans have difficulties answering.

## 2. RELATED WORK

**Question Answering.** As users are increasingly going beyond simple keyword queries, we have seen a resurgence of interest in question answering. One line of work has focused on question answering over text [6, 21, 35]. These approaches cast the user’s question into a keyword query that can be fed to a standard Web or text search engine. However, only limited attempts are made at interpreting the query. Another line of research has focused on question answering based on structured data. Early expert systems and question answering systems, such as BASEBALL [16], SHRDLU [30] and LUNAR [31], were limited to a very specific domain. While traditional systems such as LILOG [18] used extensive hand-crafted knowledge bases, modern QA systems often rely on the Web of Linked Data to support open-domain question answering. IBM’s Watson project [12] combined question answering over text and over structured data sources. However, their system does not attempt to interpret questions with complex joins.

**Query Analysis.** For query analysis, Frank et al. [13] used lexical-conceptual templates for query generation. Li et al. [22] proposed question answering for XML data by mapping questions to XQuery expressions. Unger and Cimiano [29] relied on an ontology-based natural language grammar, while Zou et al. [37] proposed a graph algorithm. Large-scale linguistic resources can also be useful for natural language understanding [25]. In contrast, the semantic parsing community has focused in part on data-driven solutions [1, 2]. The OQA [10] system adds question paraphrasing and query reformulation, mining millions of rules from a question corpus. By exploiting large amounts of data, these systems are able to account for a greater spectrum of question wordings. This increase in recall, however, comes at the expense of a decreased precision.

An alternative strategy is to exploit high-quality natural language processing models by decomposing the task as follows [17, 33]: 1) question segmentation and phrase detection, 2) mapping phrases to semantic items, 3) forming semantic triples consisting of such items, and 4) generating a SPARQL query. We follow a similar strategy, but consider the setting of both the input query and the knowledge store being multimodal.

**Multimodal Knowledge.** Our system is unique in that, unlike previous work, we consider multimodal queries against a multimodally enriched knowledge graph. Most image search engines are based on keywords (e.g., exploiting image labels and tags) or on visual image similarity. The idea of using sketches to retrieve visually similar images was proposed by [3, 4]. Their work shows that sketches can be a natural way for users to express what they have in mind. Recently, the idea of visual question answering has been proposed [36], which involves answering questions about an image, such as: *What is on the table?*, *Who is the man in this picture?* Our system, in contrast, uses image analysis techniques to help us choose answers from a knowledge base with millions of facts.

## 3. FRAMEWORK

**Overview.** We consider the setting of a user expressing an information need using a query that consists of both a natural language part and a visual part. The latter can be either a sketch or a photograph – both easy to create on mobile devices. In particular, the user has the ability to provide such images for any entity mentions in the natural language part of the query.







An overview of our system is given in Figure 2. Given the two parts of the multimodal query, our system constructs a structured

query that is used to query a large open-domain knowledge graph, which includes visual knowledge as well. Our system achieves this by detecting relevant expressions (entity mentions, classes, and relations) in the query, and mapping these onto entities, classes, and relations in the knowledge graph. At the same time, the input images are analysed and compared with images of entities in the knowledge base. A joint disambiguation is performed based on an integer-linear program with constraints pertaining to both the natural language and multimodal parts. The disambiguation results can finally be used to construct an executable SPARQL query.

**Multimodal Knowledge.** We assume the existence of a knowledge graph as exemplified in Table 1, with subject-predicate-object triples. Predicates express relations such as *means*, *type*, or *isMarriedTo*. Each relation has classes for its domain and range. The subjects and objects are concepts. These can be entities with canonical identifiers such as `Philadelphia_Phillies` and `Philadelphia_(film)` that allow us to unambiguously refer to a given entity. They can also be literals such as strings or numbers, or classes of entities, e.g. `person` or `film`.

We additionally assume that the knowledge graph contains images for entities, as is the case for Freebase, DBpedia (which now includes Wikimedia Commons), and others [9, 19, 28, 32]. Additionally, knowledge bases lacking such images can also be augmented by drawing on the Web or on large-scale image collections.

**Table 1: Excerpt of multimodal knowledge graph.**

Subject	Predict	Object
Tom_Hanks	<i>isMarriedTo</i>	Rita_Wilson
”Philadelphia”	<i>means</i>	Philadelphia_(film)
”Philadelphia”	<i>means</i>	Philadelphia_Phillies
Philadelphia_(film)	<i>type</i>	film
actor	<i>isSubclassOf</i>	person
Philadelphia_(film)	<i>hasImage</i>	
Philadelphia_(film)	<i>hasImage</i>	
Philadelphia_(film)	<i>hasImage</i>	
Philadelphia_Phillies	<i>hasImage</i>	
Philadelphia_Phillies	<i>hasImage</i>	
Philadelphia_Phillies	<i>hasImage</i>	

## 4. GRAPH CONSTRUCTION

Our first goal is to create a disambiguation multigraph capturing the choices our system will need to consider in interpreting a user query. Such a multigraph will be of the form  $G = (V, E)$ , where  $V = V_s \cup V_p$  and  $E = E^{\text{sim}} \cup E^{\text{coh}}$ . Here,  $V_s$  is the set of semantic items (*s-nodes*) in the knowledge graph,  $V_p$  is the set of natural language phrases and images (*p-nodes*),  $E^{\text{sim}} \subseteq V_p \times V_s$  is a set of weighted similarity edges that capture the strength of the mapping of a phrase/image to a semantic item, and  $E^{\text{coh}} \subseteq V_s \times V_s$  is a set of weighted coherence edges that capture the semantic coherence between two semantic items.

In the example in Fig. 3, we assume a simple user-drawn sketch of the movie poster for *Philadelphia* and an image found online for the entity name *Finley*.

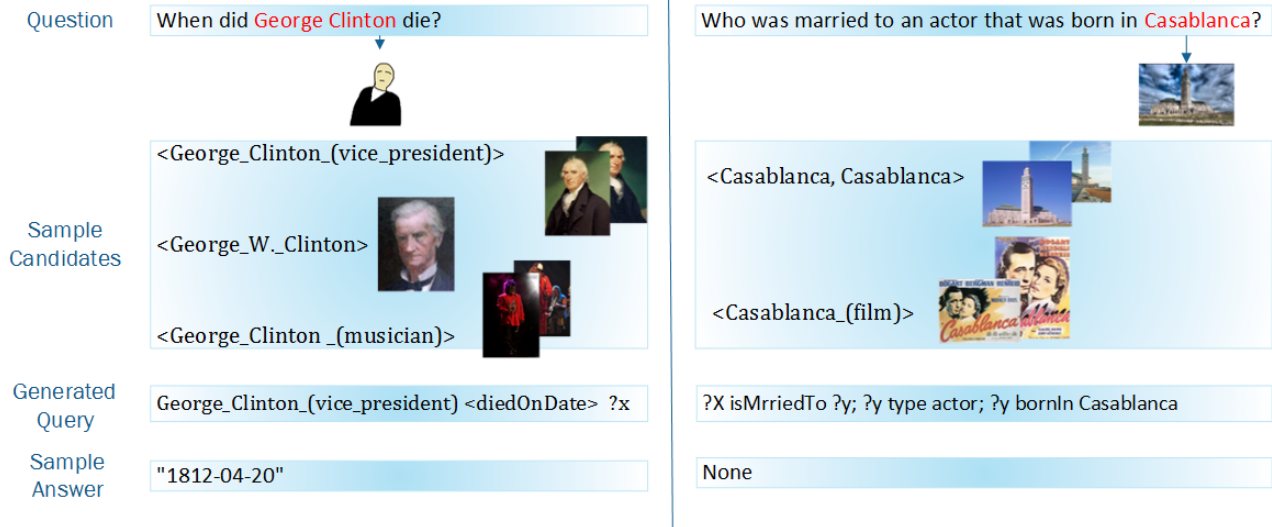


Figure 1: Sample questions and queries.

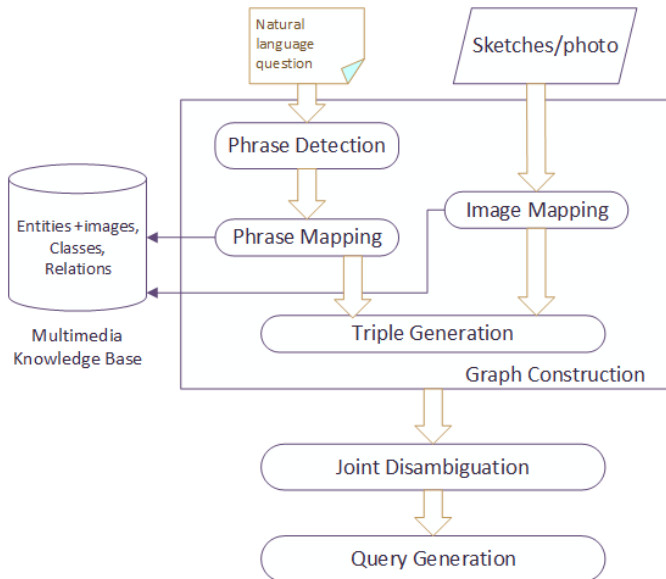


Figure 2: Workflow diagram.

## 4.1 Phrase Detection and Mapping

To construct such a graph, we begin by detecting relevant phrases within the input natural language query. Such phrases are either concept- (entity, class) or relation-denoting ones, and mapped to corresponding semantic items.

**Entities.** For entities, a named entity recognizer (NER) tool is used to detect possible references [23]. Because of its low coverage, we also directly query the knowledge base to find possible entities that a given phrase could denote, considering n-grams starting with a specific POS tag (JJ, NN, RB, or VB) as candidates. Each entity mention is thus mapped to a set of candidate entities (e.g., *Finley* to *Karen\_Finley* and *Clement\_Finley*).

**Classes.** We look up potential class names  $p$  in the knowledge graph. If phrase  $p$  is initially mapped to class  $c$  in the knowledge graph, then  $p$  is also mapped to the top- $k$  classes  $c_1, c_2, c_3$  that are similar

to  $c$  in terms of the cosine similarity of their corresponding word vectors, using Google’s word2vec skip-gram with negative sampling vectors. In our experiments, we choose  $k = 3$ . For example, *film* may be mapped to both  $c: film$  and  $c: movie$ . While this method sometimes helps in finding the right candidates, it may also introduce noise. For example, *actor* may be mapped to *Actor\_(album)*. Thus, we will later have to make sure that the algorithm makes reasonable final choices.

**Relations.** We use the pattern-to-relation table from PATTY [24], but extend its recall by directly connecting verbs to all relations they are mapped to, even if the preposition is missing. For example, *play* occurs both in the  $[[adj]] play in$  pattern (*actedIn* relation) and in  $[[adj]] play for$  (*playsForTeam* relation). Thus, we generally map *play* to both *playsForTeam* and *actedIn*. The final decision is made at the later triple generation step.

Table 2: Example of extended pattern-to-relation mappings

Pattern	Domain	PATTY pattern	Range	Relation
play	person	$[[adj]] play in$	team	<i>playsForTeam</i>
play in	person	$[[adj]] play in$	team	<i>playsForTeam</i>
play in	person	$[[adj]] play in$	movie	<i>actedIn</i>
married	person	later married to	person	<i>isMarriedTo</i>

**Image Mapping.** Our system matches input images with indexed images in the knowledge base, choosing as candidates those assigned to top candidates from the textual phrase mapping. This is based on the rationale that textual content is typically much more reliable than mere visual similarities. For similarity computation, we rely on a multitude of visual features, including GIST, HOG, SIFT and deep learning models. Details are discussed in Section 6. In the experiments, we use a threshold ( $\kappa = 0.85$ ) to filter out irrelevant query image links. If the similarity scores for all images is below  $\kappa$ , the query becomes a text-only query. For each class, there may be multiple entities, each with multiple images in the KB. Thus, for each concept (entity or class), the weights  $w_{qc}^{sim}$  for edges between query image  $M_q$  and a semantic item  $M_c$  are computed as the average similarity with all of the images above threshold  $\kappa$  for the concept  $c$ .

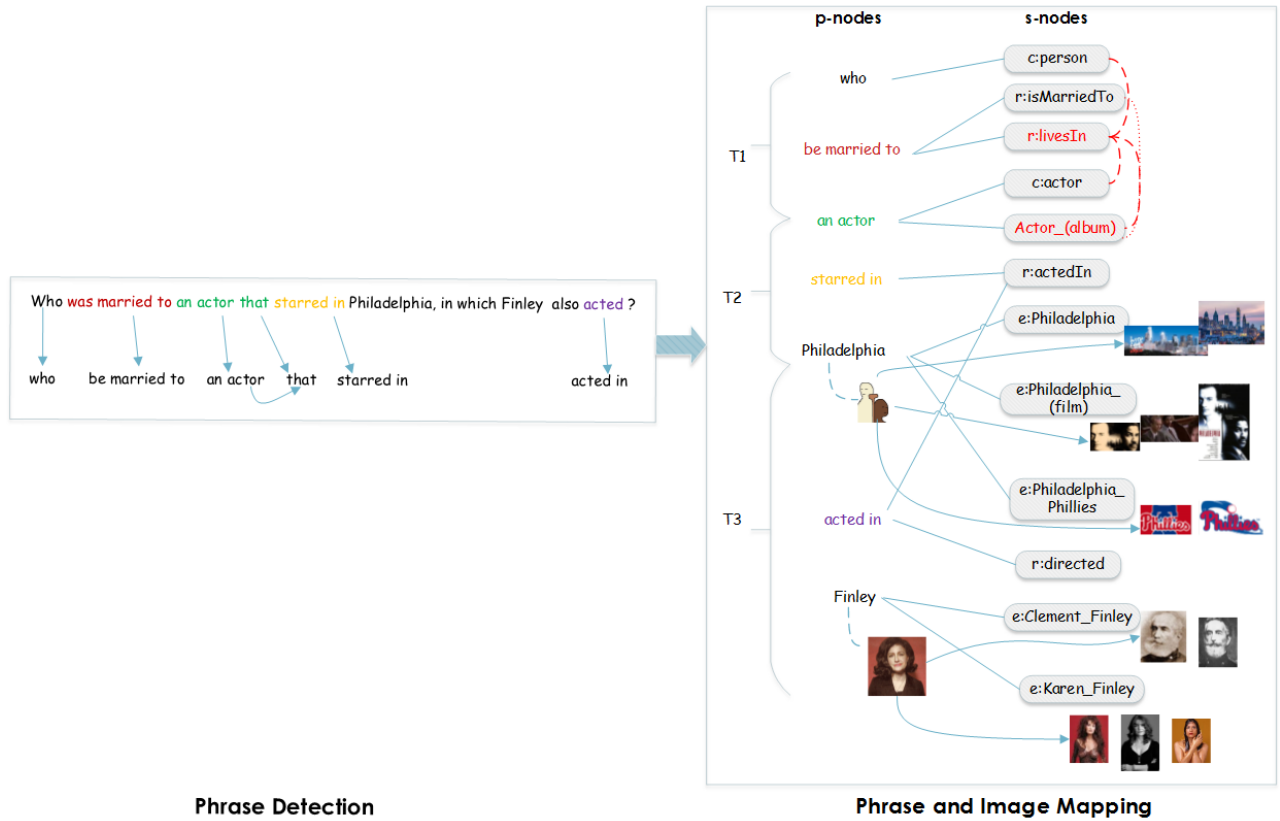


Figure 3: Example for Graph Construction. The left side shows the question, which is decomposed into phrases, while the right part illustrates the mapping of phrases to the multimodally enriched knowledge base.

## 4.2 Triple Generation

**Parse Analysis.** To generate triples, we first generate an expanded syntax tree ( $T_1$  in Figure 4) using a dependency parser [23]. We then traverse  $T_1$  to find the best matched relation phrase and its subject and object. We define  $\langle n, a_1, a_2 \rangle$  as a potential SPARQL query pattern, in which  $n$  is the relation phrase node, and  $a_1$  and  $a_2$  are two arguments. To find possible  $\langle n, a_1, a_2 \rangle$ , we choose edges in  $T_1$  with specific Stanford Dependency tags. As subject relation edges for  $a_1$ , we consider those labelled with *subj*, *nsubj*, *nsubjpass*, *csubj*, *csubjpass*, *xsubj*, or *poss*. As object relation edges for  $a_2$ , we consider: *obj*, *pobj*, *dobj*, and *iobj*.

**Relation Expansion.** We define  $C(n, a_1, a_2)$  as the coherence or similarity between  $\langle n, a_1, a_2 \rangle$  and a  $\langle \text{pattern}, \text{domain}, \text{range} \rangle$  entry in the pattern-to-relation mapping table. For the PATTY patterns, we only use the verb and the last preposition for matching with  $n$ . The verb in  $n$  is matched to the verb of a PATTY pattern using the word2vec vectors mentioned earlier. If  $n$  is composed of a verb and preposition, we add additional rules to compute exceptions. For example, “a soccer player *transferred from* a club” is different from “a soccer player *transferred to* a club”. Although both patterns have the same verb *transferred*, their similarity should be very low.

There are three kinds of nodes in  $T_1$ . Subject/object nodes refer to p-nodes with tokens mapped to a class or entity (e.g., nodes *who* and *actor*). Relation nodes refer to p-nodes with tokens mapped to a relation (e.g., node *isMarriedTo*). The earlier phrase detection and mapping produces a set of relation phrases. In Algorithm 1, for each node  $n_i$  in  $T_1$ , we check if a token in node  $n_i$  occurs in the relation phrase set. If not, we simply skip to the next node.

Otherwise, we need to merge the syntactic child node  $n'_i$  with  $n_i$ . Take *married* as an example. We use the relation edges to find the  $\langle n_i, s, o \rangle$  and compute  $C(n_i, a_1, a_2)$ , selecting the triple  $\langle \text{married to}, \text{who}, \text{actor} \rangle$  in Figure 4. Then for each child  $n'_i$  of  $n_i$  in the syntax tree, we check if  $n_i + n'_i$  (denoting a concatenation in the order given by the original sentence, with an extra space in between) is also in the relation phrase set. If so, we continue to find further children  $n''_i$ . In this case, we find  $\langle \text{be married to}, \text{who}, \text{actor} \rangle$  in Figure 4. If the new  $C(n_i + n'_i + n''_i, a_1, a_2)$  is higher than  $C(n_i, a_1, a_2)$ , we delete the previous child and use the new full concatenation to replace node  $n_i$ . Here, we replace node *married* and *married to* with *be married to*. We continue the same sort of same processing with the newly updated tree until there are no changes left to be performed. Then, we proceed to select the best mapping of a relation node in the next simplified dependency tree, i.e.,  $T_2$ , and so on.

**Argument Selection.** To find  $\langle n, a_1, a_2 \rangle$  and compute  $C(n, a_1, a_2)$  with respect to the arguments, we proceed as follows. Assuming  $n$  is a relation node, we check whether there is a subject relation between  $n$  and one of its children (using the edge labels in the dependency tree). If so, we add this child to the subject argument  $a_1$ . Likewise, the object argument  $a_2$  is recognized based on the object relations. In a dependency tree, passive voice is a special case: If there is an *agent* edge between the relation node and the object node, we swap subject and object.

Assuming a triple  $\langle n, a_1, a_2 \rangle$  has been found, this triple must be compatible with a  $\langle \text{pattern}, \text{domain}, \text{range} \rangle$  entry in the pattern-to-domain mapping table. We compute the similarity  $\mu_1$  between  $a_1$  and the domain and a second similarity  $\mu_2$  between  $a_2$  and the



range, again using the word2vec approach. Then,  $C(n, s, o) = 0.5 \times \mu_1 + 0.5 \times \mu_2$ .

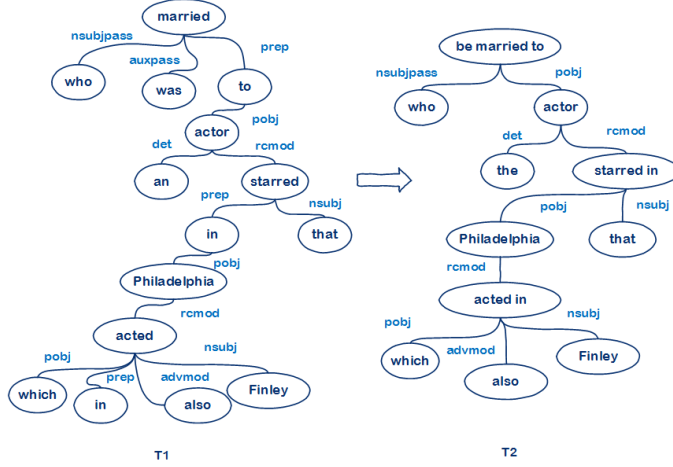


Figure 4: Dependency Tree. Left is original dependency tree; right is simplified dependency tree.

#### Algorithm 1 Triple Generation.

**Input:** Dependency tree  $T_1$ , relation phrases  $R$ .  
**Output:** Simplified dependency tree  $T_2$  and triple;  
1: Triple set  $P \leftarrow \emptyset$   
2: **for** each node  $n_i \in T_1$  **do**  
3:   **if** token in node  $n_i \in R$  **then**  
4:     Find triple  $\langle n_i, a_1, a_2 \rangle$  from  $T_1$   
5:     Compute  $C(n_i, a_1, a_2)$   
6:      $P \leftarrow P \cup \{\langle n_i, a_1, a_2 \rangle\}$   
7:     **for** each child  $n'_i$  of  $n_i$  **do**  
8:       **if**  $n'_i \in R$  **then**  
9:         Find triple  $(n_i + n'_i, a_1, a_2)$  from  $T_1$   
10:         Compute  $C(n_i + n'_i, a_1, a_2)$   
11:         **if**  $C(n_i + n'_i, a_1, a_2) > C(n_i, a_1, a_2)$  **then**  
12:             Merge  $n'_i$  with  $n_i$  into a new node in  $T_1$   
13: **return** modified  $T_1$  and  $P$ ;

### 4.3 Edge Weights

Similar to other question answering and entity linking systems [20, 33, 34], we compute edge weights for the edges between strings (p-nodes in our graph) and target disambiguations (s-nodes in our graph), as well as for edges between target disambiguations.

**Candidate Edge Weights.** Following AIDA [20], the similarity between a phrase and a candidate entity ( $w_e^{\text{sim}}$ ) is computed according to the prominence of an entity. For instance, compared with the city of Philadelphia, Philadelphia cream cheese is somewhat less frequent. We use the number of *in*-links in a repository to define the prominence of an entity. For Wikipedia-based knowledge bases, we use *in*-links in Wikipedia, while in others, we could use the triples in the knowledge base directly. It turns out that both the city and the movie Philadelphia get a higher score than Philadelphia\_(cream\_cheese). The second ingredient is based on the overlap between the phrase’s context and the candidate entity’s context. The score becomes higher as the overlap increases. For the example phrase,  $\text{Overlap}(\text{“Philadelphia”}, e:\text{Philadelphia}) > \text{Overlap}(\text{“Philadelphia”}, e:\text{Philadelphia}_{\text{(film)}})$ . As for relation and class phrases, we again use

word2vec to compute similarities between phrases and relation/class semantic items in the knowledge base.

**Coherence Edge Weights.** For coherence scores between two semantic items, i.e., weights  $w^{\text{coh}}$  for edges in  $E^{\text{coh}}$ , we rely on the Jaccard coefficient, comparing the *in*-links between two semantic items. The *in*-links are again computed either based on Wikipedia or directly in the knowledge base.

Before adding the edge to the graph, however, we ensure that the relation s-node is compatible with the corresponding argument s-node. If not, the coherence score is not computed. In Figure 3, for instance, the phrase *be married to* is mapped to relations  $r:\text{isMarriedTo}$  and  $r:\text{livesIn}$ . The s-node  $r:\text{livesIn}$  should be removed: The domain and range signature of the *livesIn* relation is (person, location), while the two arguments of the triple generated from the question both belong to  $c:\text{person}$ , which is incompatible. The phrase *actor* is mapped to the class  $c:\text{actor}$  and to the entity  $e:\text{Actor}_{\text{(album)}}$ , but the latter is removed, again because it is not compatible with the two relations  $r:\text{isMarriedTo}$  and  $r:\text{livesIn}$ . The phrase *acted in* is mapped to the relations  $r:\text{actedIn}$  and  $r:\text{directed}$ . In this case, both are kept in the graph, as they have compatible arguments.

## 5. JOINT DISAMBIGUATION

The previous steps yield a graph that encodes possible candidates for the mapping. At this point, we use all of this information to make a joint decision about the disambiguation, in particular about the mapping from phrases to semantic items. While some candidates may have already been pruned out, in most cases, the bulk of the disambiguation remains to be done, so that each phrase is assigned to at most one semantic item. In order to consider all mappings jointly and take into account the many non-trivial interdependencies and constraints involved in this, we model this problem as a mixed integer-linear program (MILP). Our overall goal, encoded in the objective function, is to select the most coherent set of mappings, which can be regarded as a subgraph of the disambiguation graph.

**Variables.** We use the following variables:

- $E_{ij} \in \{0, 1\}$  refers to the edges in  $E^{\text{sim}}$  from p-nodes for concept phrases to s-nodes for entities/classes, and  $w_{ij}^e$  denotes their respective weights.
- $M_{ij} \in \{0, 1\}$  refers to edges in  $E^{\text{sim}}$  from images for phrases to images for entities, with  $w_{ij}^m$  denoting the respective weight for the multimodal similarity.
- $R_{ij} \in \{0, 1\}$  refers to edges in  $E^{\text{sim}}$  from p-nodes for relational phrases to s-nodes for relations in the knowledge base, and  $w_{ij}^r$  denotes their respective edge weights.
- $Z_{ij} \in \{0, 1\}$  refers to edges between s-nodes, and  $w_{ij}^{\text{coh}}$  denotes the corresponding coherence weights.
- $N_i^p, N_j^s \in \{0, 1\}$  are variables indicating whether a given p-node or s-node, respectively, has been selected.
- $C_j \in \{0, 1\}$  indicates whether s-node  $j$  is a class.

**Objective.** The objective is to maximize

$$\alpha \sum_{i,j} w_{ij}^e E_{ij} + \gamma \sum_{i,j} w_{ij}^r R_{ij} + \beta \sum_{i,j} w_{ij}^m M_{ij} + \tau \sum_{i,j} w_{ij}^{\text{coh}} Z_{ij}$$

subject to the following constraints:

1.  $\forall i : \sum_j E_{ij} \leq 1$   
A given concept can only be assigned at most one corresponding entity or class.

2.  $\forall i : \sum_j R_{ij} \leq 1$   
A relational phrase can only be assigned at most one relation.
3.  $\forall i : \sum_j M_{ij} \leq 1$   
Each query image may be assigned at most one entity image in the knowledge base.
4.  $\forall i, j, i', j', I(i, i'), I(j, j') : M_{ij} > E_{i'j'}$   
The image edge  $M_{ij}$  should remain consistent with the corresponding entity edge  $E_{ij}$  ( $I(k_1, k_2)$  indicates that image node  $k_1$  is associated with p-node or s-node  $k_2$ ).
5.  $\forall i, j : E_{ij} \leq N_i^p, E_{ij} \leq N_j^s, M_{ij} \leq N_i^p, M_{ij} \leq N_j^s, R_{ij} \leq N_i^p, R_{ij} \leq N_j^s$   
The choice of edges from p-nodes to s-nodes needs to be consistent with the choice of nodes.
6.  $\forall i, j : Z_{ij} \leq \sum_k [E_{ki} + R_{ki}], Z_{ij} \leq \sum_l [E_{lj} + R_{lj}]$   
If  $Z_{ij} = 1$ , indicating that s-nodes  $i$  and  $j$  are both chosen, then there must be p-nodes mapping to each of them ( $E_{ki} = 1$  or  $R_{ki} = 1$  for some  $k$  and  $E_{lj} = 1$  or  $R_{lj} = 1$  for some  $l$ ).
7.  $\forall i, i', j, j', T(j, j') : C_j + C_{j'} \geq N_i^p + E_{ij} + N_{i'}^p + E_{i'j'} - 3$   
Each triple should have at least one class, where  $T(j, j')$  indicates that there is a triple with  $j$  as subject and  $j'$  as object, as identified by our triple detection phase (Section 4.2).

After disambiguation, each phrase is mapped to just one semantic item in the multimodal knowledge store.

**Knowledge Base Query Generation.** After this final disambiguation, we translate the triples into a SPARQL query over the knowledge base to retrieve the answers from it. For this, wh-words are replaced by variables, and every semantic class is replaced by a distinct type-constrained variable. For example, the triple `?x isMarriedTo actor` becomes `?x isMarriedTo ?y; ?y type actor`. Subsequently, the grouped triples are easily translated into an executable SPARQL query.

## 6. EXPERIMENTAL EVALUATION

### 6.1 Setup

**Dataset.** To evaluate our system, we emulated the kind of evaluation conducted in the CLEF QALD series of tasks [5], but adapted it to our setting of multimodal querying. Some questions in QALD do not contain entities at all, while others contain entity mentions that are not ambiguous, and even others contain entities that do not appear to have typically associated images. This is the case, for example, for many organizations or for certain kinds of people. We thus relied on student helpers who were instructed to emulate the sentence structures and kinds of entities used in the QALD evaluations, while coming up with a total of 20 new queries involving particularly ambiguous entity names. They were also asked to provide images for each entity. We focus on sketches created following the MindFinder approach [3], while in an additional experiment photos were chosen from Flickr in order to have regular user-generated content, though these could not be found for all entities.

For tuning, all system parameters were manually selected based on results on a very small development set of around 5 question-answer pairs. For the objective function, for instance, we use  $\alpha = \frac{3}{10}$ ,  $\beta = \frac{1}{3}$ ,  $\gamma = \frac{1}{4}$ ,  $\tau = \frac{1}{10}$ .

**Knowledge Base.** As the knowledge base for our main experiments, we used YAGO 2 [19], with over 10 million entities and 120 million facts. We added images by retrieving the top-30 Google image search results for the long version of an entity name, considering *all* disambiguation candidates for strings appearing in the benchmark query dataset.

**Table 3: Questions.**

Questions	
Q1	Who was married to the actor that starred in Philadelphia, in which Finley also acted?
Q2	Where is Paris in America?
Q3	When was Charles Bachman born?
Q4	What does Euromos mean?
Q5	Which team did Big Ben play for?
Q6	How many books did Bernstein write?
Q7	What is Grant Hill famous for?
Q8	Was John Backus born in Philadelphia?
Q9	How tall is Michael Jordan?
Q10	When did George Clinton die?
Q11	Who was called Scarface?
Q12	Which book was written by Kerouac in 1988?
Q13	Who founded Philips?
Q14	Which country does the Ganges start in?
Q15	What did a 1911 American look like?
Q16	What is the nickname of San Francisco?
Q17	Is Juliana a dog?
Q18	Which movies did Kurosawa direct?
Q19	Who created the film Beethoven?
Q20	Where does Mona Lisa live?

**Table 4: Overall evaluation in terms of accuracy**

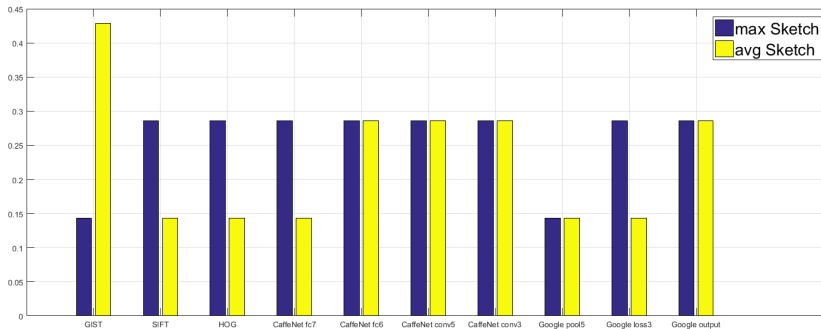
	Full	Text	Image
Question to Triples	89.3%	89.3%	89.3%
Disambiguation	78.3%	47.8%	65.2%
Final Answers	75.0%	30.0%	60.0%
Humans	10.0%		
Humans with access to Knowledge Base	90.0%		

### 6.2 Evaluation Results

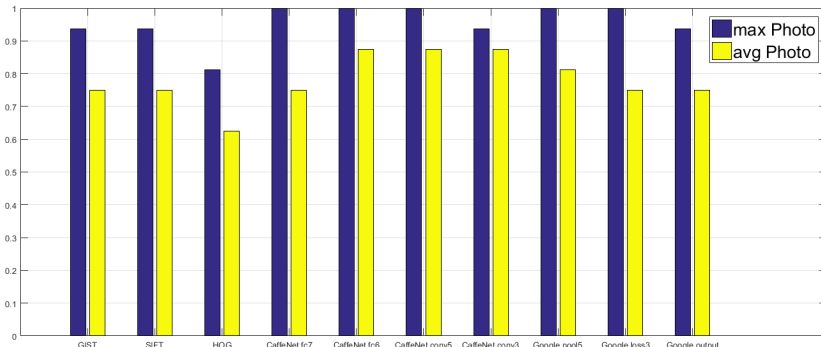
**Overall Answer Evaluation.** Table 4 provides an overall summary of the results. Most importantly, we find that the joint text+image disambiguation outperforms other systems with a correct final answer rate of 75%. For comparison, two human assessors were asked to answer all the questions in two different settings. They were first asked to provide answers without checking our knowledge base, obtaining a score of only 10% (2 correct answers) on average. Next, they were asked to answer questions with the ability to consult the knowledge base via a browsing and SPARQL interface. Here, they were able to outperform our system, getting 18 out of 20 (90%) correct. The two mistakes made by humans were because it can be hard to distinguish certain kinds of entities, e.g. people, based on mere sketches. Our system also suffers from this problem but additionally has to cope with noisy linguistic analyses and limited coverage also at the previous steps. Several steps need to work out well for us to ultimately obtain the correct answer. In the following, we study these steps in more detail.

Table 5: Evaluation of Components

	Variant	cov. (micro)	prec. (micro)	cov. (macro)	prec. (macro)
<b>Disambiguation</b>	<b>Text</b>	47.8%	50.0%	47.5%	47.5%
	<b>Image</b>	65.2%	68.2%	67.5%	67.5%
	<b>Full</b>	78.3%	81.8%	82.5%	82.5%
<b>Triple Generation</b>		89.3%	96.2%	85.0%	94.4%



(a) Sketches



(b) Photos

Figure 5: Image Features.

**Detailed Analysis.** We intercepted our system at the Triple Generation and Disambiguation stages and let humans evaluate the intermediate results. We used two judges with possible adjudication by a third one to resolve disagreements. Following [33], we report precision and coverage. Since a query can contain more than one entity, we provide both micro-averaged and macro-averaged results.

For the triple generation phase in Table 5, ideally we would obtain 28 triples and 23 entity mentions from the 20 queries. However, we only get 25 triples correct, for a micro-averaged coverage of 89.3%. The 3 incorrect triples stem from errors in the relation mapping. For example, the system does not succeed at mapping the phrase *the nickname of*, which appears in the question *What is the nickname of San Francisco?* Unfortunately, *the nickname of* was not included in our pattern-to-relation mapping table from the PATTY resource, so it could not be mapped to any relation in the knowledge base.

For the disambiguation phase, since all classes were disambiguated correctly, we only report entity disambiguation results. In Table 5, we see that the text-only disambiguation mode successfully disambiguated 10 out of 22 entities in the dataset. 12 phrases were mapped to incorrect entities when only using text disambiguation, leading to wrong answers. The *image* results refer to using only

the query images to disambiguate entity mentions, based on the image mapping procedure introduced in Section 4.1. The corresponding entity of the best-matching image (with highest  $w_{qc}^{sim}$ ) is chosen as the answer. The image-only approach got 15 correct. However, image disambiguation on its own is often insufficient. Distinguishing different people just based on an image is sometimes impossible, especially if it is just a simple sketch. For the joint system, the disambiguation rate rises to 78.3%. The answer rate also rises to  $\frac{15}{20} = 75.0\%$ . These results thus establish the effectiveness of our approach. For example, for the question *When did George Clinton die?*, the text-only disambiguation maps *George Clinton* to *George\_Henry\_Clinton*, which does not reflect the user intent for this query. The image provides additional information for the system to realize that the question aims at *George\_Clinton\_(vice\_president)*. Sometimes both text and image disambiguation are wrong, particularly if the ambiguous entities are similar in appearance. For example, different American automobiles, or even many people may look quite similar.

**Image Similarities.** We also evaluate the disambiguation by using query images a) with just black-and-white sketches, and b) color pictures from Flickr. The annotators drew black-and-white sketches

**Table 6: Comparison of disambiguation methods.**

	Correct	Wrong
<b>Full disambiguation</b>		
Question to Triples	25/28	3/28
Disambiguation	18/23	5/23
Final Answers	15/20	5/20
<b>Text-only disambiguation</b>		
Question to Triples	25/28	3/28
Disambiguation	11/23	12/23
Final Answers	6/20	14/20
<b>Image-only disambiguation</b>		
Question to Triples	25/28	3/28
Disambiguation	15/23	8/23
Final Answers	12/20	8/20

only for the seven entities for which no Flickr photos could be found. We evaluated the query images using traditional features (GIST, SIFT, HOG), as well as features from deep learning models (several different layers from the BVLC CaffeNet<sup>1</sup> and GoogleNet models [27]). As each candidate entity has multiple images, we test both the maximum and average similarity of these images for each feature.

The results are given in Table 7 and Figure 5. The deep neural models perform much better on photos, while GIST is best for sketches. In virtue of this, we generally use GIST for sketch mapping, and CaffeNet\_fc6, CaffeNet\_conv5 for photo mapping (averaging their similarities). In future work, we could easily also incorporate face recognition features.

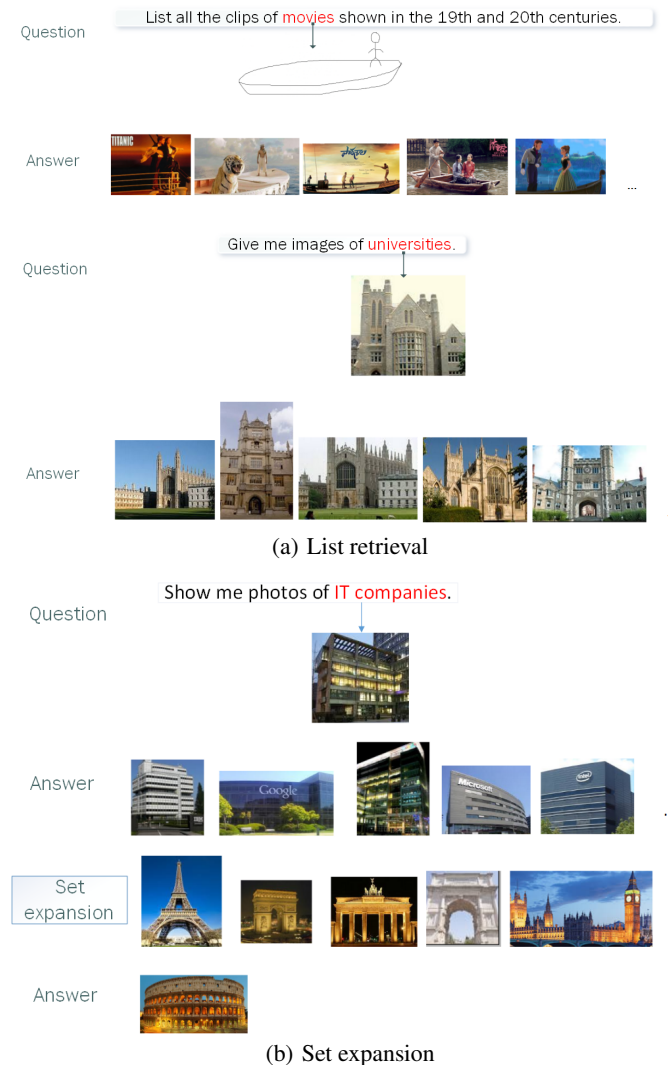
**Analysis of Text-based Disambiguation.** Additionally, we evaluated our system using text-only disambiguation, comparing it with other QA systems on the QALD-3 benchmark<sup>2</sup> over both DBpedia and YAGO. Table 8 shows that our system, when set to perform text-only disambiguation, answers the same number of questions as CASIA [17], which is more than DEANNA [33], but slightly less than [37]. There is also another system called squall2sparql [11] that is able to answer 96 questions correctly, but only if the questions are first manually rephrased in an artificial controlled language. This shows that the text-only version of our system achieves comparable results with other state-of-the-art QA systems and that the additional improvements that we obtained earlier using multimodal knowledge do not just stem from having a weak text-only baseline.

### 6.3 Discussion and Outlook

Although we have focused on ambiguous entities, our system also supports class queries. For this, we either directly associate classes with images, or recursively visit subclasses and instances until we have a set of images for the leaf entity nodes. Figure 6 presents examples of this process. Note that this goes notably beyond what current image search engines achieve. Given a query for *IT companies* with an office building as query image, a regular engine

without any type constraint would return numerous different office buildings similar to the query image, not just those of IT companies.

A related use case, also shown in Figure 6, is automatic set expansion. Given a sufficient number of input images, such as of the Eiffel Tower, Palace of Westminster, and so on, our system can automatically guess that they all belong to the same class *European capital*. It can then return related images such as of Rome, Athens, and so on, which tend to have other landmarks. Without this sort of type detection, a regular system simply returns photos that are visually similar to some of the input images. For some classes of entities,

**Figure 6: Example for list retrieval and set expansion.**

it is challenging to find typical images, e.g. for organizations, for which the images are often logos or buildings that are not particularly distinct. Locations, such as cities, are often represented by landmarks, and cities without well-known landmarks often have quite similar images of cityscapes. Thus, whether multimodal queries are beneficial may depend on the class of entities being considered.

## 7. CONCLUSION

We have presented the first approach to extend question answering over structured data to the multimodal case, allowing users to provide additional images as context to hint at their search intent.

<sup>1</sup>[https://github.com/BVLC/caffe/tree/master/models/bvlc\\_reference\\_caffenet](https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet)

<sup>2</sup><http://www.sc.cit-ec.uni-bielefeld.de/qald/>



**Table 7: Image Feature Evaluation**

	GIST	SIFT	HOG	CaffeNet_fc7	CaffeNet_fc6
<b>Photo (max)</b>	15/16	15/16	13/16	16/16	16/16
<b>Photo (avg)</b>	12/16	12/16	10/16	12/16	14/16
<b>Sketch (max)</b>	1/7	2/7	2/7	2/7	2/7
<b>Sketch (avg)</b>	3/7	1/7	1/7	1/7	2/7
	CaffeNet_conv5	CaffeNet_conv3	Google_pool5	Google_loss3	Google_output
<b>Photo (max)</b>	16/16	15/16	16/16	16/16	15/16
<b>Photo (avg)</b>	14/16	14/16	13/16	12/16	12/16
<b>Sketch (max)</b>	2/7	2/7	1/7	2/7	2/7
<b>Sketch (avg)</b>	2/7	2/7	1/7	1/7	2/7

**Table 8: Comparison of text-only QA systems.**

Ours (text)	Zou et al. 2014	DEANNA	CASIA
29	32	21	29

Our approach relies on a cascade of steps that are used to construct a graph that captures the relevant decision space. A constrained optimization problem is then solved to obtain the final disambiguation and generate a knowledge base query to retrieve the answer.

Not only can such multimodal querying in some cases be a more natural way of conveying one’s search intent, perhaps based on mental imagery. Our results also demonstrate that a system’s accuracy can increase substantially when jointly considering the natural language and image parts of the query. Our work thus makes important inroads towards exploring the space of multimodal forms of knowledge seeking.

One direction for further research is to improve the natural language capabilities of our system so as to support a more diverse range of questions [7], and incorporating dialogue capabilities. Another direction involves improved fine-grained classification of images and video obtained by drawing on massive amounts of user-generated content as training data [8, 14, 26]. With improved models to detect specific entities such as the Guggenheim Museum in New York, our system would be able to better narrow down the candidate set even when the textual query is very unspecific. A third research direction is to explore means of presenting the retrieval results visually [15].

Overall, given the enormous explosion of online media content, paired with the ubiquity of mobile devices and the rapid growth of augmented reality, we believe that in the future there will be an increasing need for further research on multimodal information and knowledge management.

## 8. ACKNOWLEDGMENTS

We would like to thank all reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China under Grant (No. 61503217), National 973 Program (No. 2015CB352500), the Joint NSFC-ISF Research Program 61561146397, jointly funded by the National Natural Science Foundation of China and the Israel Science Foundation, the China National Key Research and Development Project (2016YFB1001403), and the Shandong Provincial Science and Technology Development Program (2016GGX106001).

## References

- [1] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. In *The Conference on Empirical Methods on Natural Language Processing*, pages 1533–1544, 2013.
- [2] Q. Cai and A. Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *The annual meeting of the Association for Computational Linguistics (1)*, pages 423–433. Citeseer, 2013.
- [3] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang. MindFinder: interactive sketch-based image search on millions of images. In *ACM Multimedia Conference*, pages 1605–1608. ACM, 2010.
- [4] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2Photo: Internet image montage. *ACM Transactions on Graphics (TOG)*, 28(5):124, 2009.
- [5] P. Cimiano, V. Lopez, C. Unger, E. Cabrio, A.-C. Ngonga Ngomo, and S. Walter. Multilingual question answering over linked data (qald-3): Lab overview. In *Proceedings of CLEF 2013*, pages 321–332. Springer Berlin Heidelberg, 2013.
- [6] H. T. Dang, D. Kelly, and J. J. Lin. Overview of the TREC 2007 question answering track. In *TREC*, volume 7, page 63. Citeseer, 2007.
- [7] G. de Melo and K. Hose. Searching the web of data. In *Proc. ECIR 2013*, LNCS. Springer, 2013.
- [8] G. de Melo and N. Tandon. Seeing is believing: The quest for multimodal knowledge. *ACM SIGWEB Newsletter*, (Spring 2016), 2016. ISSN 1931-1745. URL <http://dl.acm.org/citation.cfm?id=2903517>.
- [9] G. de Melo and G. Weikum. MENTA: Inducing multilingual taxonomies from Wikipedia. In J. Huang, N. Koudas, G. Jones, X. Wu, K. Collins-Thompson, and A. An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM 2010)*, pages 1099–1108, New York, NY, USA, October 2010. ACM. ISBN 978-1-4503-0099-5.
- [10] A. Fader, L. Zettlemoyer, and O. Etzioni. Open question answering over curated and extracted knowledge bases. In *ACM SIGKDD Conferences on Knowledge Discovery and Data Mining*, pages 1156–1165. ACM, 2014.
- [11] S. Ferré. squall2sparql: a translator from Controlled English to full SPARQL 1.1. In *QALD-3r*, 2013.

- [12] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3):59–79, 2010.
- [13] A. Frank, H.-U. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer. Question answering from structured knowledge sources. *Journal of Applied Logic*, 5(1):20–48, 2007.
- [14] C. Gan, M. Lin, Y. Yang, G. de Melo, and A. G. Hauptmann. Concepts not alone: Exploring pairwise relationships for zero-shot video activity recognition. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*. AAAI Press, 2016.
- [15] T. Ge, Y. Wang, G. de Melo, and H. Li. Visualizing and curating knowledge graphs over time and space. In *Proceedings of ACL 2016*. ACL, 2016.
- [16] B. F. Green, Jr., A. K. Wolf, C. Chomsky, and K. Laughery. Baseball: An automatic question-answering. In *Western Joint IRE-AIEE-ACM, IRE-AIEE-ACM '61 (Western)*, pages 219–224, 1961.
- [17] S. He, K. Liu, Y. Zhang, L. Xu, and J. Zhao. Question answering over linked data using first-order logic. In *The 2017 Conference on Empirical Methods on Natural Language Processing*, 2014.
- [18] O. Herzog, J. H. Siekmann, and C. Rollinger. *Text Understanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence*. Springer-Verlag, 1991.
- [19] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. De Melo, and G. Weikum. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *The World Wide Web conference*. ACM, 2011.
- [20] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenu, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *The 2017 Conference on Empirical Methods on Natural Language Processing*, 2011.
- [21] C. Kwok, O. Etzioni, and D. S. Weld. Scaling question answering to the web. *ACM Transactions on Information Systems*, 19(3):242–262, 2001.
- [22] Y. Li, H. Yang, and H. Jagadish. NaLIX: A generic natural language search environment for XML data. *ACM Transactions on Database Systems*, 32(4):30, 2007.
- [23] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *The annual meeting of the Association for Computational Linguistics*, 2014.
- [24] N. Nakashole, G. Weikum, and F. Suchanek. PATTY: a taxonomy of relational patterns with semantic types. In *ENMLP and CoNLL*, pages 1135–1145. ACL, 2012.
- [25] J. Rouces, G. de Melo, and K. Hose. FrameBase: Representing n-ary relations using semantic frames. In *Proceedings of ESWC 2015*, pages 505–521, 2015.
- [26] E. Shutova, N. Tandon, and G. de Melo. Perceptually grounded selectional preferences. In *Proceedings of ACL 2015*, pages 950–960, 2015.
- [27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [28] N. Tandon, G. de Melo, A. De, and G. Weikum. Knowlywood: Mining activity knowledge from Hollywood narratives. In *Proceedings of CIKM 2015*, 2015.
- [29] C. Unger and P. Cimiano. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *International Conference on Natural Language and Information Systems*, pages 153–160. Springer, 2011.
- [30] T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.
- [31] W. A. Woods and R. Kaplan. Lunar rocks in natural English: Explorations in natural language question answering. *Linguistic structures processing*, 5:521–569, 1977.
- [32] H. Xu, Y. Wang, K. Feng, G. de Melo, W. Wu, A. Sharf, and B. Chen. Shapelearner: Towards shape-based visual knowledge harvesting. In G. A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum, and F. van Harmelen, editors, *Proceedings of ECAI 2016*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 435–443. IOS Press, 2016. ISBN 978-1-61499-671-2. doi: 10.3233/978-1-61499-672-9-435.
- [33] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum. Natural language questions for the web of data. In *The Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning*, pages 379–390, 2012.
- [34] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum. AIDA: An online tool for accurate disambiguation of named entities in text and tables. *International Conference on Very Large Data Bases*, 4(12):1450–1453, 2011.
- [35] Z. Zheng. AnswerBus question answering system. In *HTL*, pages 399–404, 2002.
- [36] Y. Zhu, C. Zhang, C. Ré, and L. Fei-Fei. Building a large-scale multimodal knowledge base for visual question answering. *arXiv 1507.05670*, 2015.
- [37] L. Zou, R. Huang, H. Wang, J. X. Yu, W. He, and D. Zhao. Natural language question answering over RDF: a graph data driven approach. In *Proc. ACM-SIGMOD Int. Conf. on Management of Data*, pages 313–324. ACM, 2014.