# Knowledge Engineering via Human Computation

**Elena Simperl, Maribel Acosta, and Fabian Flöck**

## What Is Knowledge Engineering

Knowledge engineering refers to processes, methods, and tools by which knowledge in a given domain is elicited, captured, organized, and used in a system or application scenario (Studer et al. 1998). The resulting 'knowledge base' defines and formalizes the kinds of things that can be talked about in that particular context. It is commonly divided into a 'schema', also called 'ontology', and the actual data the application system manipulates. The data is described, stored, and managed as instantiations of the concepts and relationships defined in the ontology. With applications in fields such as knowledge management, information retrieval, natural language processing, eCommerce, information integration or the emerging Semantic Web, ontologies were introduced to computer science as part of a new approach to building intelligent information systems (Fensel 2001): they were intended to provide knowledge engineers with reusable pieces of declarative knowledge, which can be together with problem-solving methods and reasoning services easily assembled to high-quality and cost-effective systems (Neches et al. 1991; Schreiber et al. 1999). According to this idea, ontologies are understood as shared, formal domain conceptualizations; from a system engineering point of view, this component is strictly separated from the software implementation and can be thus efficiently reused across multiple applications (Guarino 1998).

The emergence of the Semantic Web has marked an important stage in the evolution of knowledge-driven technologies. Primarily introduced by Tim Berners-Lee

E. Simperl (✉)
Web and Internet Science Group, University of Southampton, Southampton, UK
e-mail: e.simperl@soton.ac.uk

M. Acosta • F. Flöck
Karlsruhe Institute of Technology, Institute AIFB, Karlsruhe, Germany
e-mail: maribel.acosta@kit.edu; fabian.floeck@kit.edu

P. Michelucci (ed.), *Handbook of Human Computation*,
DOI 10.1007/978-1-4614-8806-4_13, 

(2001), the idea of providing the current Web with a computer-processable knowledge infrastructure in addition to its original, semi-formal and human-understandable content foresees the usage of knowledge components which can be easily integrated into and exchanged among arbitrary software environments. In this context, the underlying knowledge bases are formalized using Web-based, but at the same time semantically unambiguous representation languages that are pervasively accessible and can (at least theoretically) be shared and reused across the World Wide Web. Although the combination of human-based computation and Semantic Web technologies yields promising results,[1] the implementation of such hybrid systems raises a whole set of new challenges which are discussed in detail in the chapter 'The Semantic Web and the Next Generation of Human Computation' of this book.

As a field, knowledge engineering is mainly concerned with the principles, processes, and methods that produce knowledge models that match this vision. It includes aspects related to knowledge acquisition, as a key pre-requisite for the here identification and organization of expert knowledge in a structured, machine-processable way, but also to software engineering, in particular when it comes to the actual process models and their operationalization. Last, but not least, knowledge engineering has strong ties to artificial intelligence and knowledge representation, in order to translate the results of the knowledge elicitation phase into structures that can be reasoned upon and used in an application system. In addition, it shares commonalities with several other areas concerned with the creation of models to enable information management, including entity relationships diagrams in relational data base systems engineering, and object-oriented programming, UML and model-driven architectures in software engineering. Each of these areas defines their specific way to capture domain knowledge, represent and exploit its meaning in the creation of innovative systems and applications.

In this chapter, we will look into several important activities in knowledge engineering, which have been the subject of human computation research. For each activity we will explain how human computation services can be used to complement existing automatic techniques, and give a short overview of the state of the art in terms of successful examples of systems and platforms which showcase the benefits of the general idea.

## Why and Where Is Human Computation Needed

Many aspects of the knowledge engineering life cycle remain heavily human-driven (Siorpaes and Simperl 2010). Prominent examples include, at the technical level, the development of conceptualizations and their use in semantic annotation, the evaluation and curation of knowledge resources, the alignment of ontologies and data integration, as well as specific types of query processing and question answering

[1] Especially in tasks like data annotation or data quality assessment, which involve defining and encoding the meaning of the resources published on the Web or resolving semantic conflicts such as data ambiguity or inconsistency.

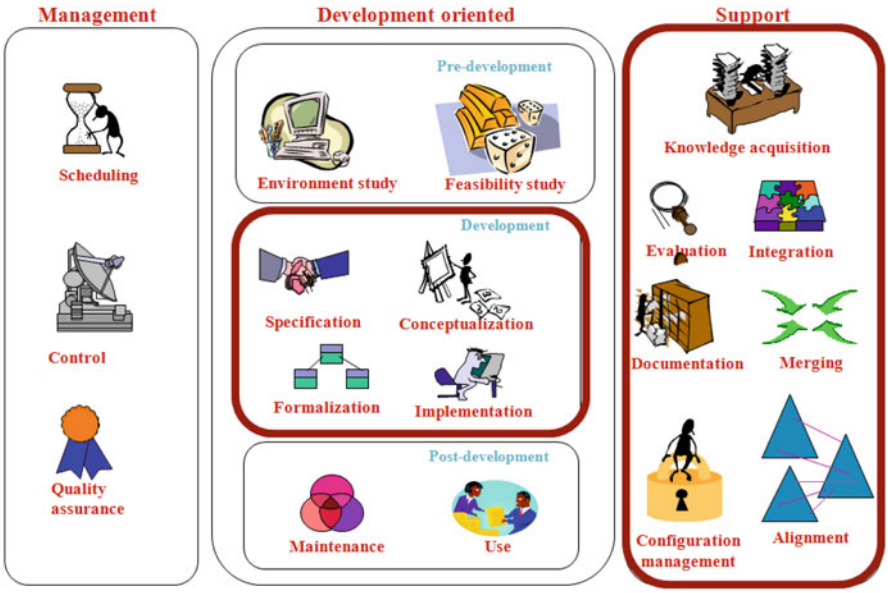


**Fig. 1** Ontology engineering activities (Gómez-Pérez et al. 2004)

(Maribel Acosta et al. 2012). To an equal extent, it comprises almost everything that has to do with the creation of human-readable interfaces to such sources, in particular labeling, where human capabilities are indispensable to tackle those particular aspects that are acknowledged to be hardly approachable in a systematic, engineering-driven fashion; and also, though to a lesser extent, to the wide array of methods and techniques that have been proposed as an attempt to perform others automatically. In this second category, despite constant progress in improving the performance of the corresponding algorithms and the quality of their results, experiences show that human assistance is nevertheless required, even if it is just for the validation of algorithm outputs.

Figure 1 gives an overview of the knowledge engineering life cycle at the level of ontologies.[2] The activities we will look into are an essential part of knowledge (or ontology) engineering, but are not necessarily unique to this area. Nevertheless, compared to software engineering or relational data bases, it is primarily knowledge engineering, and in particular its use on the (Semantic) Web, that has been increasingly the subject to a crowdsourcing approach. This is due primarily to the (recent) strong Web orientation of knowledge engineering as a field, which led to a variety

[2] A similar process model applies to the creation, management and use of instance data. Management and pre-development activities cover the entire scope of the knowledge-engineering exercise. Development, post-development and support activities are equally relevant to both schema and data, though there might be differences in their actual realization. For example, instance data is typically lifted from existing sources into the newly created ontological schema, while a greater share of activities at the ontology level are carried out manually.

of knowledge base development projects and applications thereof being initiated and executed with the help of open Web communities, leveraging Web 2.0 participatory principles and tools. The high costs associated with creating and maintaining real-world knowledge bases in a classical work environment motivated experiments with alternative approaches that rely on the wisdom of open crowds, volunteer contributions, or services such as Amazon Mechanical Turk. Especially the latter is still an expanding field of research, with initial trials for various types of knowledge domains and tasks delivering very promising results. Nevertheless, in scenarios which are less open, both in terms of audiences addressed and the technologies they use, crowdsourcing methods need to take into account additional aspects to be effective, including the human and computational resources available, and how the results could be optimally acquired from, and integrated into, productive environments while avoiding to disrupt established workflows and practices.

The development life cycle in Fig. 1 distinguishes among management, development, and support activities. *Ontology management* refers primarily to scheduling, controlling and quality assurance. Scheduling is about coordinating and managing an ontology development project, including resource and time management. Controlling ensures that the scheduled tasks are accomplished as planned. Finally, quality assurance evaluates the quality of the outcomes of each activity, most notably of the implemented ontology. *Ontology development* can be split into three phases: pre-development, development, and post-development. As part of the pre-development phase, an environment study investigates the intended purpose and use of the ontology. Furthermore, a feasibility study ensures that the ontology can actually be built within the time and resources assigned to the project. These two activities are followed by the actual development, which includes first and foremost the requirements specification that eventually results in a conceptual model and its implementation in a given knowledge representation language. In the final, post-development phase, the ontology is updated and maintained as required; this phase also includes the reuse of the ontology in other application scenarios. *Support* stands for a wide range of different activities that can be performed in parallel or subsequent to the actual ontology development. The aim of these activities is to augment the results of the, typically manual, ontology development by automatizing parts of the process, providing auxiliary information sources that could be used to inform the conceptualization and implementation tasks, and evaluating and documenting intermediary results. Typical support activities include *knowledge acquisition*, *ontology evaluation*, *ontology alignment*, and *ontology learning* and *ontology population*. *Ontology population* is closely related to *semantic annotation*, by which information artifacts of various forms and flavors are described through instances of a given ontology. *Data interlinking* is closely related to the area of *ontology alignment*, and involves the definition of correspondences between entities located in different data sets, and the description of these correspondences through specific predicates (equivalence, related to, or domain-specific ones). The two activities not only share commonalities in terms of the types of basic (machine-driven) algorithms they make use of, but can also influence each other. Based on mappings at the schema level, one can identify potentially related instances; conversely, the availability of links between sets of entities may indicate similarities between classes.

In previous work of ours (Siorpaes and Simperl 2010) we surveyed methodologies, methods and tools covering each activity in order to learn about the types of processes knowledge and ontology engineering projects conform to, and the extent and reasons they might rely on human intervention. In the remainder of this section, we summarize the results of this analysis for a selection of activities: conceptual modeling as part of ontology development, alignment and interlinking as a prominent support activity in the engineering life cycle introduced earlier, and finally documentation, as a classical example of human-driven activity.

## *Developing Ontologies*

Developing ontologies requires domain expertise and the ability to capture domain knowledge in a clean but purposeful conceptual model. An ontology describes the things that are important in a specific domain of interest, their properties, and the way they are interrelated. It defines a common vocabulary and the meaning of the terms used in the vocabulary. In the last 15 years, a wide array of ontology development methodologies have been proposed (Gómez-Pérez et al. 2004). Many suggest to start with the specification of the scope the ontology should cover and the requirements it should fulfil. This is often complemented by the informal and formal specification of competency questions. Based on that, relevant terms in the domain are then collected. Widely accepted ontology representation formalisms use classes, properties, instances and axioms as ontological primitives to describe domain knowledge. The overall process can be performed in a centralized (within a predefined team of knowledge engineers and domain experts) or a decentralized fashion (within a potentially open community of stakeholders, domain experts, and users).

The *conceptual modeling* process includes the definition of classes and the associated class hierarchy, as well as the definition of properties and additional axioms. Several automatic approaches have been proposed to discover specific types of relationships, in particular specialization and generalization extracted from natural language text, but human intervention is required for training the underlying algorithms, building the text corpus on which they operate, and validating their results (Bouquet et al. 2006; Buitelaar and Cimiano 2008). In addition, efforts need to be typically invested in post-processing the domain and ranges of individual properties, so that these are defined at the most appropriate level in the abstraction hierarchy. Defining axioms, on the other side, involves specifying precise, logics-based rules, such as cardinality constraints on certain properties and disjointness that apply to classes. Approaches for automatically specifying such axioms are very limited in their scope and require substantial training and validation (Völker et al. 2007).

As explained previously, the creation of instances is related to *semantic annotation*; we investigate it in more detail below. Relevant for the context of ontology development is the definition of so-called ‘fixed’ or ‘ontological’ instances which are the result of explicit modeling choices during the conceptualization phase. The distinction between classes and instances is very specific to the application setting, and we are not aware of any approaches aiming at automatizing this task.

There is a wide range of approaches that carry out *semi-automatic annotation of texts*: most of them make use of natural language processing and information extraction techniques. Even though they require training, a large share of the work can be automated (Reeve and Han 2005; Uren et al. 2006). The situation is slightly different with the *annotation of multimedia* content: approaches for the annotation of media, no matter if manual, semi-automatic or automatic, aim at closing the so-called "semantic gap", which is a term coined to describe the discrepancy between low-level technical features of multimedia, which can be automatically extracted to a great extent, and the high-level, meaning-bearing features a user is typically interested in and refers to when searching for content. Recent research in the area of semantic multimedia retrieval attempts to automatically derive meaning from low-level features, or other available basic metadata. This can so far be achieved to a very limited extent, i.e., by applying machine learning techniques with a vertical focus for a specific domain (such as face recognition), in turn for a substantial training and tuning, all undertaken with human intervention (Bloehdorn et al. 2005). The *annotation of Web services* is currently a manual task, but more research is needed in order to clearly determine whether this can be traced back to the nature of the task, or to the fact that the corresponding area is not mature enough to produce approaches that can offer reliable automatic results (Dimitrov et al. 2007; Kerrigan et al. 2008, 2007).

In Siorpaes and Simperl (2010) we analyzed various tools for text and media annotation which create semantic metadata with respect to the degree of automation they can support (nine tools in the first category, and six in the second one). In the case of textual resources, the main challenge is finding optimal ways to integrate human inputs (both workflow-wise and implementation-wise) with existing pre-computed results. On the contrary, multimedia annotation remains largely unsolved; there the typical scenario would use human computation as a main source of input for the creation of annotations, though specific optimizations of the process are nevertheless required. In Simperl et al. (2013) we embark on a broader discussion about how users could be motivated to engage with different types of participatory applications, including human computation ones, and on the principles and methods that could be applied to study and change user behavior to encourage engagement.

## *Supporting Ontology Development*

Support activities accompany the development of ontologies. One prominent example thereof is the *alignment* of heterogeneous ontologies. Many of the existing ontology engineering environments provide means for the manual definition of mappings between ontologies. In addition, there is a wide range of algorithms that provide automatic support (Euzenat et al. 2007; Euzenat and Shvaiko 2007; Noy and Musen 2001, 2003), whilst it is generally accepted that the question of which ontological primitives match cannot (yet) be done fully automatically (Euzenat and Shvaiko 2007; Falconer and Storey 2007). This area is closely related to *data interlinking*, which we analyzed in more detail in (Simperl et al. 2012; Wölger et al. 2011).

Another support task is *documentation*, which contains two main components: the documentation of the overall process, and of the resulting knowledge base, in particular in terms of labels and commentaries associated to concepts, attributes, properties, axioms, and instances of the knowledge base. Either way, it remains human-driven, especially when it comes to recording modeling decisions and their rationales. Basic support for ontology documentation can be obtained by automatically creating entries for each ontological primitive which capture its core context in terms of labels and other annotations, as well as related classes, instances and properties. In this context, it is also worth mentioning the topic of *ontology localization*, which mainly refers to the translation of labels into different natural languages. Similarly to other areas in ontology engineering which employ natural language processing techniques for instance, ontology learning human input is required in order to solve translation questions which are highly context-specific, or to choose between different alternative translations.

We now turn to an analysis of how human computation could be applied to these activities in order to overcome the limitations of automatic techniques. For each activity, we will introduce examples of systems and applications such as games-with-a-purpose, microtask crowdsourcing projects, and community-driven collaborative initiatives that demonstrate the general feasibility of a hybrid human-machine approach.

## Games-with-a-Purpose for Knowledge Engineering

Games-with-a-purpose is one of the most popular instances of social computing approaches to knowledge acquisition proposed in the last years. The game designer capitalizes on the appeal of key game properties such as fun, intellectual challenge, competition, and social status to turn the significant number of hours willingly spent playing by users through sophisticated algorithms into meaningful results that lead to qualitative improvements of information management technology. The concept is particularly useful to problems in knowledge engineering, an area which historically has targeted highly specialized audiences rather than casual Internet users. Tasks such identifying classes as groups of similar individuals, relating objects through properties, defining sub- and super-classes, validating whether two entities are the same or different, or labeling things in a given natural language are, though not always trivial to answer, much easier to tackle by humans than by machines.[3]

In the remainder of this section we will give a number of examples for this type of games illustrating the general concept.

[3] Exceptions include highly contextualized systems, which require extensive training and/or background knowledge. In these cases, the manual efforts shifts from the creation and maintenance of the knowledge base to the generation of training data sets and background corpora.

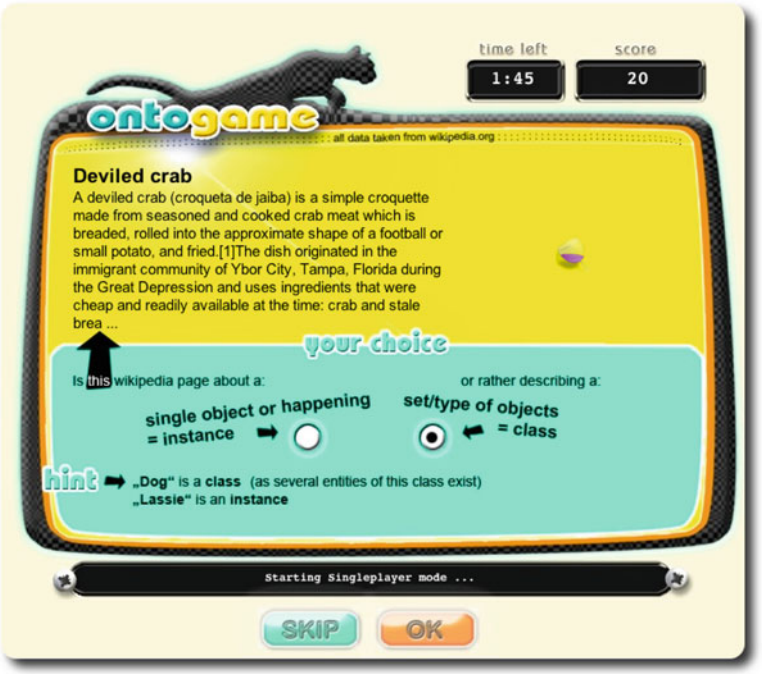


**Fig. 2** OntoPronto: Expanding an existing ontology with Wikipedia concepts

## *Conceptual Modeling*

### OntoPronto

OntoPronto (Siorpaes and Hepp 2008) (see Fig. 2) is a real-time quiz game for the development and population of ontologies. The knowledge corpus used to generate challenges to be addressed by players is based on the English Wikipedia. Random Wikipedia articles are classified to the most specific class of an upper-level ontological structured called Proton (SEKT). The game can be played in a single- and two-players modus, where the former uses pre-recorded answers to simulate interaction. In the most general case, two players are randomly playing and can gain points by consensually answering two types of questions referring to the same Wikipedia article. In the first step, they are shown the first paragraph of an article and (if applicable) a picture, and are asked to agree whether the topic of the article stands for a class of similar objects or a concrete object. Once this issue has been settled, they enter the second step of the game, in which they navigate through the hierarchy of classes of the Proton ontology in order to identify the most specific level which will be extended through the topic represented by the Wikipedia article. The game back-end uses a number of standard means to validate the players' results.


**Fig. 3** Virtual Pet: Creating a Chinese semantic network

Questions are subject to several game rounds and repeated, consensual answers are considered correct if they were authored by reliable players.

### Virtual Pet and Rapport

Virtual Pet Game[4] (see Fig. 3) aims at constructing a semantic network that encodes common knowledge (a Chinese ConceptNet[5] equivalent). The game is built on top of PPT, a popular Chinese Bulletin Board System, which is accessible through a terminal interface. Each player has a pet which he should take care of in order to satisfy its needs, otherwise it could die. In order to take care of the pet (e.g., buy food), the player has to earn points by answering quiz-like questions that are relevant to the semantic network creation task at hand. The pet, in this game, is just a substitute for other players which receive the questions/answers and respond or validate them. Question and answers are provided by players using given templates (e.g., subject, predetermined relation, object). The validation of players' inputs is based on majority decision.

The purpose of the Rapport Game (Yen-ling Kuo et al. 2009) is very similar. Rapport Game, however, is built on top of Facebook (see Fig. 4) and uses direct interaction between players, rather than relying on the pet-mediated model

[4] http://agents.csie.ntu.edu.tw/commonsense/cate2_1_en.html

[5] http://conceptnet5.media.mit.edu/


**Fig. 4** Rapport Game: Building a semantic network via Facebook

implemented by Virtual Pet. The players ask their opponents questions. These, in turn, answer them and the answers are evaluated by the user community. Points are granted for each type of action, from raising questions to answering and rating.

### Guess What?!

Guess What?![6] (see Fig. 5) is a semantic game-with-a-purpose that creates formal domain ontologies from Linked Open Data.[7] Each game session is based on a seed concept that is chosen manually. In the back-end the application tries then to find a matching URI in a set of pre-defined ontologies of the Linking Open Data Cloud and gather additional information about the resources identified by the URI from interconnected Linked Data repositories. Additional information relies mainly on the adjacent graph in the Linking Open Data Cloud, including related classes and

[6] http://nitemaster.de/guesswhat/manual.html

[7] http://linkeddata.org/

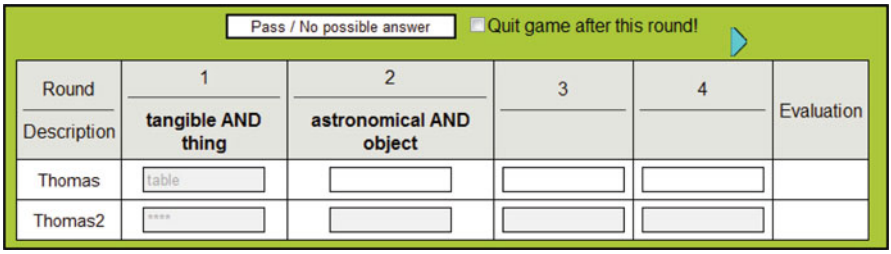


**Fig. 5** Guess What?!: identifying complex concepts

entities, but also documentation such as labels. The resulting labels and URIs are analyzed using natural language processing techniques in order to identify expressions which can be translated into logical connectors such as 'AND'. Complex descriptions are broken down into smaller fragments, which are then weighed by a generality and confidence value. These fragments are used to generate the challenges solved in each game round. More specifically, a round starts with the most general fragment, and in the subsequent rounds a more specific one is connected to it through a logical operator. The goal of each round is for the player to guess the concept described by the interconnected fragments. For instance, in an initial round the fragment shown to the user contains the fragments 'fruit' AND 'yellow' AND 'oval', with solutions such as 'lemon' OR 'citrus'. Quality assurance is achieved through consensus and majority voting.

## *Alignment and Interlinking*

### WordHunger

WordHunger[8] (see Fig. 6) is a turn-based Web application that integrates among two large knowledge bases: WordNet and Freebase.[9] WordNet is a large lexical data base in which elements are grouped into synonym sets. Freebase is a structured knowledge base. Each game round consists of a WordNet term and up to three suggested possible Freebase concepts. The player then has to select the most fitting of those, or in case of insecurity, pass. Players may also select "no match" in case the articles are not related. After one of these possible choices the player proceeds to the next WordNet term. A player gets a point for each answer given. The data is validated through repeated answers.

[8] http://wordhunger.freebaseapps.com/

[9] WordNet: http://wordnet.princeton.edu/, Freebase: http://www.freebase.com/

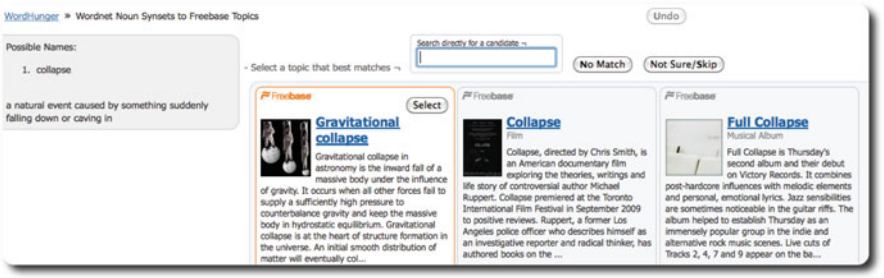


**Fig. 6** WordHunger: Mapping WordNet to Freebase


**Fig. 7** SpotTheLink: ontology alignment illustrated on DBpedia and Proton

## SpotTheLink

SpotTheLink (Siorpaes and Hepp 2008) (see Fig. 7) is a real-time quiz-like game for ontology alignment. It aligns random concepts of the DBpedia ontology[10] to the Proton upper-level ontology that was already used in OntoPronto. Each game round

[10] http://wiki.dbpedia.org/Ontology


**Fig. 8** UrbanMatch: connecting points of interest with image data sets

is centered around a randomly chosen DBpedia concept. In the first step of a game round, both players have to select a fitting concept from the Proton ontology. In case they choose the same concept they proceed with agreeing on a relationship between these concepts, either *is the same* or *is more specific*. They earn points for each consensual answer. After successfully matching a DBpedia class with a Proton class the players have to match the same DBpedia to the hierarchical next level of the Proton class. Otherwise, they play a new random DBpedia class. The validation of the results is based on consensus and majority voting.

### UrbanMatch

UrbanMatch[11] is an application used to interlink Smart Cities data sources by exploiting games-with-a-purpose and gamification techniques (Fig. 8). It is built as a mobile, location-aware application in which players are expected to match points of interest related to a urban area to representative photos retrieved from the Web. To generate the challenges to be played, in each game round the application uses a mixture of trusted and less trusted online sources, including OpenStreeMap,[12] a geo-information repository, Flickr and Wikimedia Commons, the collection of images used by

[11] http://swa.cefriel.it/urbangames/urbanmatch/index.html

[12] http://www.openstreetmap.org

the Wikipedia encyclopedia. Candidate links are validated based on a metric taking into account the source of the image and of the answers. There are six difficulty levels and two game modes: in a 'race against time' players maximize the number of links founds between points of interest and pictures, thus optimizing recall; accuracy is addressed by a 'wise choice' option in which players have to identify the best possible links and submit their best-four selection without any time constraints.

### *Semantic Annotation*

There is a wide array of games applied to tasks related to object identification and annotation of multimedia content. A selection of some of these games published in the human computation literature of the last five years can be found on the SemanticGames site.[13] They apply a large variety of games models (input agreement, output agreement, see GWAP),[14] and further distinguish themselves in the choice of game narrative, quality assurance (majority voting and beyond), and selection of challenges in each game round.

## Microtask Crowdsourcing for Knowledge Engineering

In this section we introduce a number of approaches that have used microtask crowdsourcing to execute knowledge engineering tasks in a highly parallel fashion by using services of established crowdsourcing labor markets such as Amazon Mechanical Turk (AMT) and CrowdFlower.[15] For this purpose the actual task was first decomposed into small work units (denomintaed *microtasks*) and published on these platforms. The input collected from the crowds was incorporated into knowledge-based systems to be further consumed by the systems themselves, other automatic approaches, or even processed by human workers in more complex tasks.

### *Conceptual Modeling*

#### CrowdSPARQL: Ontological Classification

CrowdSPARQL (Maribel Acosta et al. 2012) is a hybrid query engine for graph-based data which combines automatic query processing capabilities with microtask

[13] http://www.semanticgames.org

[14] http://www.gwap.com/

[15] Amazon Mechanical Turk: http://mturk.com, CrowdFlower: http://crowdflower.com

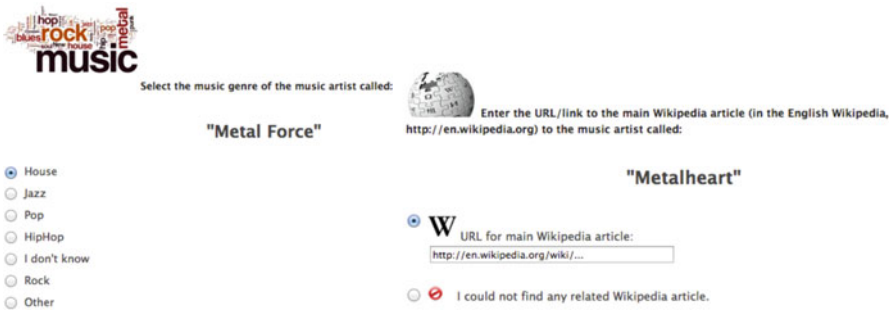


**Fig. 9** CrowdSPARQL HIT interface: Ontological classification (*left*), entity resolution (*right*)

crowdsourcing. The aim is to produce enhanced results by evaluating the SPARQL queries against data stores and crowdsourcing parts of the query to discover relationships between Linked Data resources via a microtask platform such as AMT. The human tasks created by the engine are declaratively described in terms of input and output, which allows translating the results from the crowd directly into data that can be further processed by a conventional graph query engine.

From a knowledge engineering point of view, this hybrid engine will select specific patterns in the 'WHERE' clause of a SPARQL query that refer to tasks such as ontological classification and interlinking. Where such information is not available in the original data repositories, these patterns will be translated into microtasks (see Fig. 9). CrowdSPARQL implements several mechanisms for spam detection and quality assessment, including the creation of control questions within the microtasks where the correct answer is a priori known. The new relationships provided by the crowd are evaluated using majority voting (and some variations of this rule) and the consolidated answers are integrated into the Linked Data sets.

### InPhO System: Conceptual Herarchies

The InPhO system (Niepert et al. 2007) attempts to dynamically generate a taxonomy of philosophical concepts defined in the Indiana Philosophy ontology.[16] The system relies on a user community composed of domain experts to construct and develop a philosophical hierarchy via asynchronous feedback, where the users (dis)confirm the existence of semantic relationships between the ontology concepts. The system follows a human-computation-based approach, where the feedback is collected and incorporated automatically into the taxonomy, evolving it and allowing new users' contributions.

Eckert et al. (2010) applied microtask crowdsourcing to populate the InPhO taxonomy via AMT, and compared the quality of the AMT workers' input with the

[16] https://inpho.cogs.indiana.edu/

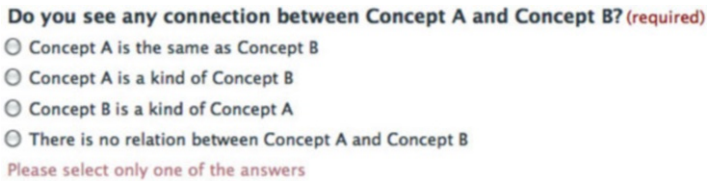


**Fig. 10** CrowdMAP human task interface

feedback provided by the experts. The experiment involved the crowdsourcing of 1,154 pairs of philosophical concepts; each HIT submitted to AMT consisted of 12 questions where the users must first determine the relatedness ('unrelated' vs. 'highly related') of concept pairs and then select a predefined semantic relationship between these concepts. Each HIT was answered by five distinct workers. In addition, the authors implemented filtering mechanisms to detect low quality answers. By applying the right combination of these filters, the results suggest that it is possible to achieve high quality answers via crowdsourcing, as the feedback from the crowd and the experts is comparable.

## *Alignment and Interlinking*

### CrowdMAP: Ontology Alignment

CrowdMAP (Sarasua et al. 2012) introduces a human-loop in the ontology alignment process by crowdsourcing the possible mappings between ontologies as microtasks with individual alignment questions. The CrowdMAP architecture receives as input two ontologies to be aligned and an automatic algorithm to generate an initial mapping. Based on this information, CrowdMAP generates the human tasks (see Fig. 10) and submits them to CrowdFlower, where the workers suggest the type relationships between a pair of concepts ('same', 'subclass of', 'superclass of'). During the microtask generation, control questions where included in the tasks in order to facilitate the spam detection. In addition, the quality assurance and answer consolidation mechanisms supported by CrowdMap are those offered by the platform CrowdFlower. The experimental study in Sarasua et al. (2012) showed that CrowdMap on average is able to outperform automatic solutions, and the results suggest that the combination of ontology alignment algorithms with human-driven approaches may produce optimal results.

### CrowdSPARQL: Entity Resolution

In Linked Data it is often the case that different data sets create their own resource identifier to refer to the same concepts. CrowdSPARQL (Maribel Acosta et al. 2012)

is designed to handle entity resolution tasks via crowdsourcing when links between data sets are required while processing a SPARQL query. The current status of the engine allows the interlinking of Linked Data resources to DBpedia, which contains the RDF representations of knowledge extracted from Wikipedia. The workers perform the discovery of 'same as' correspondences providing the Wikipedia entry (URL) for a given Linked Data resource (see Fig. 9).

#### ZenCrowd: Entity Linking

ZenCrowd (Demartini et al. 2012) is a hybrid system that combines algorithmic and manual techniques in order to improve the quality of entity extraction on a corpus of news articles and linking them to Linked Data resources, by executing state-of-the-art solutions to find candidate matches and selecting the right one via microtask crowdsourcing. In each microtask, the workers have to select the correct Linked Data resource for a given entity. The results from the crowd are analyzed by ZenCrowd using a quality model to select the right answer based on probabilistic graphs, where entities, workers and candidate matches are represented as nodes, which are connected through factors. The experimental results showed that ZenCrowd is able to outperform automatic approaches by crowdsourcing entity linking, reflected as an improvement of the overall system accuracy.

### *Documentation*

#### Mechanical Protégé: Ontology Documentation

Mechanical Protégé[17] is a plug-in for the open source Protégé[18] ontology editor tool and knowledge-base framework, which allows crowdsourcing ontology development activities such as creating classification hierarchies or labeling concepts and translating them into different languages. The ontology editor selects a task and the concepts within the ontology subject to crowdsourcing as illustrated in Fig. 11, and Mechanical Protégé creates and submits the human tasks to AMT. The types of tasks handled by Mechanical Protégé are considered complex tasks due to the variety of answers that may be retrieved from the crowd, therefore the ontology editor must perform the analysis and validation of the human input manually.

[17] http://people.aifb.kit.edu/mac/mechanicalProtege

[18] http://protege.stanford.edu/

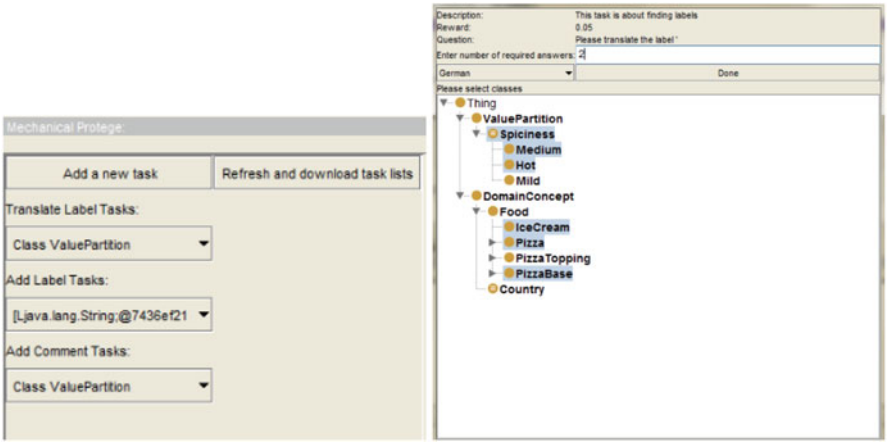


**Fig. 11** Mechanical Protégé creation of microtasks: Selecting the type of task (*left*), selecting entities to crowdsource (*right*)

## Other Approaches

Other human computation-based approaches rely on collaborative contributions from users to achieve their goals. One recent example comes from the Linked Data community for evaluating the well-known data set DBpedia. The DBpedia Evaluation Campaign,[19] aimed at detecting possible quality issues in the DBpedia data set; it was performed in two phases: first, a taxonomy of common quality issues was built by experts; then, Linked Data enthusiasts were invited to use the TripleCheckMate tool (see Fig. 12) in order to arbitrarily explore the data set resources and identify possible quality problems contemplated in the taxonomy. The second phase was performed as an open contest; the user submissions were analyzed and verified by experts, who selected a winner based on his contributions. Although the campaign has finished already, the information collected from the participants represents a valuable input to correct future versions of the data set and implement better (semi-)automatic data extractors on top of the Wikipedia mappings.[20]

While DBpedia is the attempt to extract structured, semantic data from the only partly ordered, enormous knowledge base that is Wikipedia, the project Wikidata[21] takes a different, more fundamental approach by letting the community directly build structured data relations to be then used by automated systems. This happens,

[19] http://nl.dbpedia.org:8080/TripleCheckMate/

[20] Wikipedia extractors. http://wiki.dbpedia.org/DeveloperDocumentation/Extractor

[21] http://www.wikidata.org/


**Fig. 12** TripleCheckMate tool for exploring resources and selecting quality problems

e.g., through inline queries from Wikipedia, pulling for example up to date inhabitant numbers from Wikidata into info boxes of articles about cities. Community members, mostly Wikipedia editors, establish and maintain the data entries in a collaborative, open fashion. Data entries are stored as triples and can be accessed using Linked Data technologies. Wikidata is operated by the Wikimedia Foundation as the 'Data Layer' for its projects, much like Wikimedia Commons acts as its overall storage for media files. The project bears many similarities with initiatives such as Freebase, which applied a combination of volunteer and paid crowdsourcing to collaboratively create and maintain a structured knowledge base.[22]

## Conclusions

In this chapter we gave an overview of how human computation methods such as paid microtasks and games-with-a-purpose could be used to advance the state of the art in knowledge engineering research, and develop and curate valuable (structured)

[22] http://www.freebase.com/

knowledge bases in different domains. Given the inherently human-driven nature of many knowledge engineering tasks, most notably knowledge acquisition and modeling, human computation has received great attention in this community as an alternative to costly, tedious expert-driven approaches followed in the past, with promising results. This resulted in an impressive number of systems, in particular casual games, tackling tasks as diverse as ontological classification, labeling, property elicitation, entity linking, ontology alignment or the annotation of different types of media. Besides these promising prospects, many of these projects still need to prove themselves in terms of sustainability and actual added value in the data they produce. More research is needed in order to enable the reuse of human-computation data, and even allow for different methods to be applied in combination. This would not only increase the quality of the crowd-engineering knowledge, which will be curated in time through various tools and platforms, but it would possibly facilitate the application of human-computation methods to different types of tasks and workflows, that are less amenable to parallelization.

## References

Acosta M, Simperl E, Flöck F, Norton B (2012) A sparql engine for crowdsourcing query processing using microtasks – technical report. Institute AIFB, KIT, Karlsruhe

Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. Sci Am 284(5):34–43

Bloehdorn S, Petridis K, Saathoff C, Simou N, Tzouvaras V, Avrithis Y, Handschuh S, Kompatsiaris Y, Staab S, Strintzis MG (2005) Semantic annotation of images and videos for multimedia analysis. LNCS. Springer The Semantic Web: Research and Applications, volume 3532 of Lecture Notes in Computer Science, pp 592–607. Springer Berlin Heidel-berg

Bouquet P, Scefor S, Serafini L, Zanobini S (2006) Booststrapping semantics on the web: meaning elicitation from schemas. In: 15th international conference on ACM, New York, NY, USA, http://doi.acm.org/10.1145/1135777.1135851, pp 505–512

Buitelaar P, Cimiano P (2008) Ontology learning and population: bridging the gap between text and knowledge. IOS-Press, Amsterdam

Celino, Irene, Simone Contessa, Marta Corubolo, Daniele Dell'Aglio, Emanuele Della Valle, Stefano Fumeo, Thorsten Krüger, and Thorsten Krüger. "UrbanMatch-linking and improving Smart Cities Data." In LDOW. 2012

Demartini G, Difallah DE, Cudré-Mauroux P (2012) Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st international conference on world wide web, WWW'12, Lyon. ACM, New York, pp 469–478

Dimitrov M, Simov A, Momtchev V, Konstantinov M (2007) Wsmo studio – a semantic web services modelling environment for wsmo (system description). In: European semantic web conference (ESWC 2007), Innsbruck

Eckert K, Niepert M, Niemann C, Buckner C, Allen C, Stuckenschmidt H (2010) Crowdsourcing the assembly of concept hierarchies. In: Proceedings of the 10th annual joint conference on digital libraries, JCDL '10, Gold Coast. ACM, New York, pp 139–148

Euzenat J, Shvaiko P (2007) Ontology matching. Springer, Berlin/New York

Euzenat J, Mocan A, Scharffe F (2007) Ontology alignments. Volume 6 of semantic web and beyond. Springer, p 350

Falconer SM, Storey M-A (2007) A cognitive support framework for ontology mapping. In: Asian semantic web conference (ASWC 2007), Busan

Fensel D (2001) Ontologies: a silver bullet for knowledge management and electronic commerce. Springer, Berlin/New York

Gómez-Pérez A, Fernández-Lopéz M, Corcho O (2004) Ontological engineering – with examples form the areas of knowledge management, e-Commerce and the semantic web. Advanced information and knowledge processing. Springer

Guarino N (1998) Formal ontology and information systems. In: Proceedings of the 1st international conference on formal ontologies in information systems FOIS1998, Amsterdam/ Washington. IOS-Press, pp 3–15

Kerrigan M, Mocan A, Tanler M, Fensel D (2007) The web service modeling toolkit – an integrated development environment for semantic web services (system description). In: European semantic web conference (ESWC 2007), Innsbruck

Kerrigan M, Mocan A, Simperl E, Fensel D (2008) Modeling semantic web services with the web service modeling toolkit. Technical report, Semantic Technology Institute (STI)

Kuo Y-L, Lee J-C, Chiang K-Y, Wang R, Shen E, Chan C-W, Hsu J-Y (2009) Community-based game design: experiments on social games for commonsense data collection. In: International conference on knowledge discovery and data mining, HCOMP'09, Paris. ACM, New York, pp 15–22

Neches R, Fikes RE, Finin T, Gruber TR, Senator T, Swartout WR (1991) Enabling technology for knowledge sharing. AI Mag 12(3):35–56

Niepert M, Buckner C, Allen C (2007) A dynamic ontology for a dynamic reference work. In: Proceedings of the 7th ACM/IEEE-CS joint conference on digital libraries, JCDL '07, Vancouver. ACM, New York, pp 288–297

Noy N, Musen M (2001) Anchor-prompt: using non-logical context for semantic matching. In: IJCAI workshop on ontologies and information sharing, Seattle, pp 63–70

Noy NF, Musen M (2003) The prompt suite: interactive tools for ontology merging and mapping. Int J Hum Comput Stud 59(6):983–1024

Reeve L, Han H (2005) Survey of semantic annotation platforms. ACM Press, New York, pp 1634–1638

Sarasua C, Simperl E, Noy NF (2012) Crowdmap: crowdsourcing ontology alignment with microtasks. In: International Semantic Web Conference (1)'12, Boston. pp 525–541

Schreiber G, Akkermans H, Anjewierden A, de Hoog R, Shadbolt N, Van de Velde W, Wielinga B (1999) Knowledge engineering and management: the CommonKADS methodology. MIT, Cambridge http://proton.semanticweb.org

SEKT Consortium. Proton ontology

Simperl E, Wölger S, Norton B, Thaler S, Bürger T (2012) Combining human and computational intelligence: the case of data interlinking tools. Int J Metadata Semant Ontol 7.2 (2012):77–92

Simperl E, Cuel R, Stein M (2013) Incentive-Centric semantic web application engineering. Synthesis lectures on the semantic web: theory and technology. Morgan & Claypool, San Rafael

Siorpaes K, Hepp M (2008) Ontogame: weaving the semantic web by online games The Semantic Web: Research and Applications, volume 5021 of Lecture Notes in Computer Science, pp 751–766. Springer Berlin Heidelberg, 2008

Siorpaes K, Simperl E (2010) Human intelligence in the process of semantic content creation. World Wide Web J 13(1):33–59

Studer R, Benjamins VR, Fensel D (1998) Knowledge engineering principles and methods. Data Knowl Eng 25(1/2):161–197

Uren V, Cimiano P, Iria J, Handschuh S, Vargas-Vera M, Motta E, Ciravegna F (2006) Semantic annotation for knowledge management: requirements and a survey of the state of the art. Web Semant Sci Serv Agents World Wide Web 4(1):14–28

Völker J, Vrandecic D, Sure Y, Hotho A (2007) Learning disjointness. In: Proceedings of the 4th european semantic web conference ESWC 2012, Innsbruck, pp 175–189

Wölger S, Siorpaes K, Bürger T, Simperl E, Thaler S, Hofer C (2011) Interlinking data – approaches and tools. Technical report, STI Innsbruck, University of Innsbruck

Yen-ling Kuo et al. (2009): In Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '09)