CrowdTruth: Machine-Human Computation Framework for Harnessing Disagreement in Gathering Annotated Data

Oana Inel[1,3], Khalid Khamkham[1,3], Tatiana Cristea[1,3], Anca Dumitrache[1,3], Arne Rutjes[2], Jelle van der Ploeg[2], Lukasz Romaszko[1,3], Lora Aroyo[1], and Robert-Jan Sips[3]

[1] VU University Amsterdam
k.khamkham@gmail.com, {anca.dumitrache,oana.inel,lora.aroyo}@vu.nl,
tatiana.cristea@student.vu.nl, lukasz.romaszko@gmail.com
[2] IBM Services Center Benelux, The Netherlands
arne.rutjesISC@nl.ibm.com, j.van.der.ploegISC@nl.ibm.com
[3] CAS Benelux, IBM Netherlands
Robert-Jan.Sips@nl.ibm.com

Abstract. In this paper we introduce the *CrowdTruth* open-source software framework for machine-human computation, that implements a novel approach to gathering human annotation data for a variety of media (e.g. text, image, video). The CrowdTruth approach embodied in the software captures human semantics through a pipeline of four processes: *a*) combining various machine processing of media in order to better understand the input content and optimize its suitability for micro-tasks, thus optimize the time and cost of the crowdsourcing process; *b*) providing reusable human-computing task templates to collect the maximum diversity in the human interpretation, thus collect richer human semantics; *c*) implementing 'disagreement metrics', i.e. *CrowdTruth metrics*, to support deep analysis of the quality and semantics of the crowdsourcing data; and *d*) providing an interface to support data and results visualization. Instead of the traditional inter-annotator agreement, we use their disagreement as a useful signal to evaluate the data quality, ambiguity and vagueness. We demonstrate the applicability and robustness of this approach to a variety of problems across multiple domains. Moreover, we show the advantages of using open standards and the extensibility of the framework with new data modalities and annotation tasks.

Keywords: crowdsourcing, gold standard data, machine-human computation, data analysis, experiment replication, information extraction

1 Introduction

The unprecedented amount of information available on the Web in terms of text, images and videos opens incredible opportunities and challenges for machines to interpret such data adequately. Machines are typically good in handling massive scale, e.g. indexing huge amounts of data and humans in interpreting text, images

and audio-visual content. Automated approaches for semantic interpretation are typically founded on a very simple notion of truth, while in reality the principled approach is that truth is not universal and is strongly influenced by human perspectives and the quality of the sources.

The Semantic Web had already made a huge leap by adding both diversity and machine-readable semantics of data on the Web. However, the scale of the Web provides unlimited amounts of new perspectives and interpretation contexts. Using crowdsourcing platforms such as CrowdFlower[4] or Amazon Mechanical Turk[5] (MTurk) for gathering human interpretation on data has become now a mainstream process. In the NLP field [1], crowdsourcing has been used for nearly a decade, as the low level language understanding tasks map well into micro-tasks. In the AI field [2], this has become a scalable way to gather a cheaper annotated data for gold standards that is used to train and evaluate machine learning systems. However, as we have observed previously [3], the introduction of crowdsourcing has not fundamentally changed the way gold standards are created: humans are still asked to provide a semantic interpretation of some data, with the explicit assumption that there is *one correct interpretation*. Thus, the diversity of interpretation and perspectives is still not taken in consideration.

In previous work, we have introduced the *CrowdTruth methodology*, a *novel approach for gathering annotated data from the crowd*. Inspired by the simple intuition that human interpretation is subjective [4], and by the observation that disagreement is a natural product of having multiple people performing annotation tasks, this methodology can provide useful insights about the task, a particular annotation, or a worker. We proposed rejecting the traditional notion of ground truth in gold standard annotation, in which annotation tasks are viewed as having a single correct answer, and adopting instead a disagreement-based crowd truth [5]. In [4, 6–8] we have validated *CrowdTruth* in the context of measuring the quality of workers, annotation units, and tasks. We showed experimental evidence that these measures are inter-dependent, and that existing crowdsourcing approaches that measure only worker quality are missing important information, as not all the annotated units are created equal.

This paper presents the open-source *CrowdTruth software framework* that implements the CrowdTruth methodology in a machine-human computing workflow for collecting, processing and evaluating crowdsourcing data. In this workflow, the capacities of both humans and machines are optimally combined for the output of high quality gold standard for machines to learn from. Such framework can be helpful to the Semantic Web community considering the growing number of crowdsourcing applications in this field, as well as the growing need for gold standard training and evaluation data. Significant benefits brought up by the CrowdTruth framework over the current state-of-the-art crowdsourcing frameworks such as CrowdLang [9] and Jabberwocky [10] are the deeper analysis of the annotated data and the data visualization tools. In contrast to GATECrowd [11], the presented framework has the advantage of manipulating a variety of

[4] `https://crowdflower.com/`

[5] `https://www.mturk.com/mturk/`

input media types. Moreover, the added value of the framework is increased due to the PROV[12] model integration. Thus, its generic and domain-agnostic features are essential inside CrowdTruth, as they offer a straightforward solution to (1) visualize the entire process cycle of a media unit, (2) assess the clarity of a media unit as well as (3) replicate the same process for different other media units. The open source CrowdTruth framework is available for download at `https://github.com/laroyo/CrowdTruth`, the service at `http://crowdtruth.org` and documentation as `http://crowdtruth.org/info`.

2 CrowdTruth Use Cases

Before diving into the CrowdTruth framework and its components in Section 5, we introduce the use cases in the context of which the system has been developed and tested. To ensure data diversity, each use case introduces either a new domain, content modality or a new annotation task. All the data and the experiments can be viewed in *CrowdTruth* through the *Media* section. New content can be inserted for immediate execution of new experiments through the *Upload Media* option, as described in Section 4. Below we describe the four use cases:

- IBM Watson *medical text* annotation for *factor span extraction* (FactSpan) and *relation extraction* (RelEx)
- IBM Watson *newspapers text* annotation for *event extraction* (MRP-Events)
- Sound & Vision *video* annotation for *event extraction* (NISV-Events)
- Rijksmuseum *image* annotation for *flower names extraction* (Rijks-Flowers)

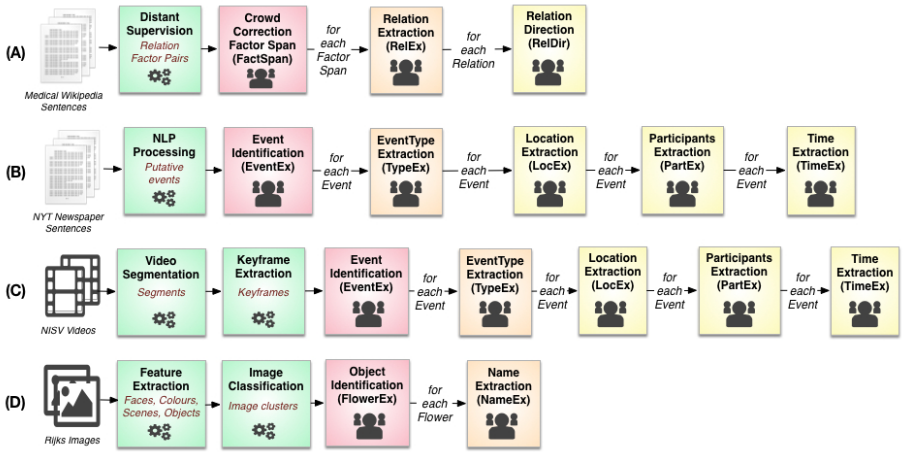

Fig. 1: CrowdTruth Annotation Workflows for Text, Images and Videos

The main experiments initiating the implementation of this framework were focussed on providing gold standard to the IBM Watson system for relation and factor extraction in medical texts. Thus, the best illustration on how the CrowdTruth Framework works can be currently observed in the *RelEx* and *FactSpan* use cases. For this, we have defined (as depicted in Fig. 1) workflow A, where medical sentences are shown to the crowd for annotation in three micro-

tasks. In the context of the MRP project at IBM, we have also experimented with newspaper text and annotations for event and named entities extraction (workflow B). Workflows C and D, show the annotation tasks on Rijksmuseum Amsterdam images and Sound & Vision videos we have performed within the context of two research projects. In the following section, Section 3, we provide a detailed description of the annotation tasks for all use cases.

3 CrowdTruth Annotation Tasks

The CrowdTruth use cases introduce about 14 distinct annotation templates across three content modalities (text, image, video) and three domains (medical, news, culture). Each of those templates has also a number of variations, depending on the target result quality. Ultimately, CrowdTruth framework is aimed to provide its template collection as a continuously extendible *library of annotation task templates*, which can be reused and adapted for new data and use cases. The implementation of CrowdTruth does not pose restrictions for the creation of new templates. To see more detailed description for all tasks and their templates, visit this page: `http://crowdtruth.org/templates/examples`. The templates themselves are accessible through the *Jobs* section in *CrowdTruth*, by selecting the *Create New Job* option. Depending on the type of content chosen, only the applicable sub-set of templates will be presented.

3.1 Medical Text Annotation: IBM Watson Medical Use Cases

- **FactSpan: Factor Span Correction**. The crowd is given a *sentence* with two highlighted $factors$ (either a word or a word phrase). For each factor, the crowd is asked to determine whether it is complete. If it is not, the workers highlight the words in the sentence that would complete the factor.
- **RelEx: Relation Identification**. The crowd is given a *sentence* with two highlighted $factors$ and a set of 12 target *relation types*. The crowd is asked to select all the relations expressed in the sentence between the given factors.
- **RelDir: Relation Direction Identification**. The crowd is given the output of $RelEx$ - a *sentence*, two highlighted $factors$, and a *relation* between the factors - and are asked to choose the direction of the relation. Since this is an easy task, we use golden units (instances with known answers - e.g. "*Aspirin* treats *headaches*") to decrease the spam rate. The advantage of this method is that CrowdFlower immediately rejects untrustworthy workers.
- **RelExDir: Relation & Direction Identification**. The crowd is given the combined task of relation and direction identification on the $FactSpan$ output. As with $RelEx$, the crowd is shown a *sentences* with two highlighted $factors$ and is asked to check all the relations that apply between them. The relations set contains the initial 12 relations and their inverses (23 in total).

3.2 Newspaper Text Annotation: IBM Watson MRP Use Case

- **EventEx: Event and Event Type Identification**. The crowd is given a *sentence* with a highlighted *putative event* (word phrase that could potentially express an event, i.e. verbs or nominalized verbs) and is asked whether it refers to an event. For each event the crowd is asked to choose the event type expressed in the sentence from an $EventType$ taxonomy (see Table 1).

– **LocEx, TimeEx, PartEx: Event Location, Participants & Time Identification**. The crowd is given a *sentence* with a highlighted *event* from the *EventEx* output, and is asked (1) to indicate whether the sentence contains *location*, *time* or *participant* for this *event*, (2) to highlight the words in text that refer to those and (3) to select their types (see Table 1).

Table 1: Event Role Fillers Taxonomies

Role Filler	Taxonomy
Event	Purpose, Arriving or Departing, Motion, Communication, Usage, Judgment, Leadership, Success or Failure, Sending or Receiving, Action, Attack, Political
Location	Geographical (Continent, Country, City); Land Area (Island, Mountain, Beach); Water Area (Ocean, River, Lake, Sea); Road/Railroad (Road, Street, Railroad); Building (Educational, Government, Residence, Commercial, Industrial, Military, Religious)
Period	Before, During, After, Repetitive, Timestamp, Date, Year, Week, Day, Part of Day
Participant	Person, Organization, Geographical Region, Nation, Object

3.3 Image Annotation: Rijksmuseum Amsterdam Use Case

– **FlowerEx: Depicted Flower Identification with Bounding Box**. In the pre-processing we identify the images with the highest chance of depicting flowers. We ask the crowd to identify all the flowers in them (by surrounding each flower with a box), and to fill in their names, the total number of flowers and the number of different flower types depicted.

3.4 Video Annotation: Sound & Vision Use Case

– **DescEventEx: Event Identification in Video Description**. The named entities are extracted during pre-processing form the video description text. The crowd is asked to confirm or reject any machine annotations on this text, and highlight all the events and their role fillers.

– **VidEventEx: Event Identification in Video**. The crowd is given a video or a video segment and is asked to annotate events that are *depicted* (literally mentioned) or *associated* (related to some spoken events/role fillers).

4 CrowdTruth Data Model

Essential to maintaining all the data resulting from the annotation tasks in Section 2 is the definition of a data model, which complies with three main requirements: (1) to be abstract enough to store different content modalities, i.e. text, images, videos, (2) to be specific enough, i.e. semi-structured, to still be able to query the data, and (3) to capture the provenance of the data. The MongoDB[6] document-oriented NoSQL database does not rely on predefined schemas, rather the structure of the data stored can be defined dynamically at any point in time. Such flexibility is a key requirement because when collecting crowdsourcing data, we often do not know upfront the appropriate structure. An example of this are the various online content processing APIs that return results in a JSON format but with different structures. MongoDB allows us to store any of these JSON results in documents without any conversion because of its BSON storage design. However, storing data without defining structure makes it difficult to query. Thus, we defined a data model that is abstract enough to be able to store any type of data, yet specific enough to be able to query this data (Figure 2).

[6] `http://www.mongodb.org/`

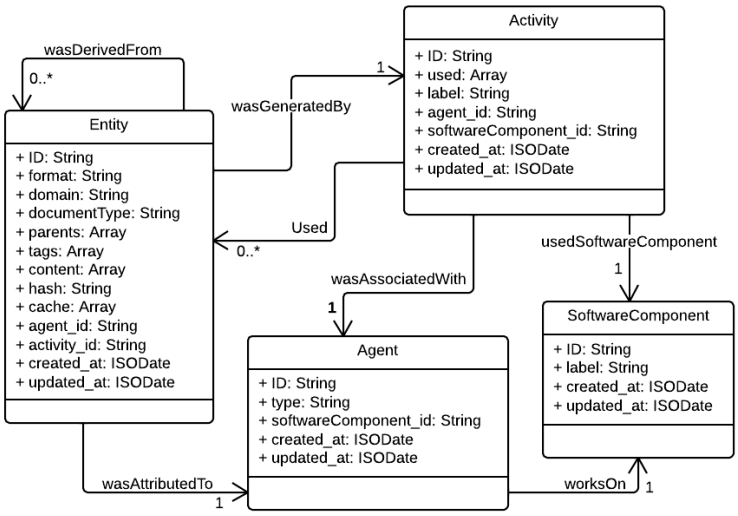

Fig. 2: The CrowdTruth Data Model and Data Provenance

The CrowdTruth MongoDB deployment hosts one database, with four collections `Entities`, `Activities`, `Agents` and `SoftwareComponents`. For every collection we define `Models` in the framework which map to their respective collections. The models are used by the Moloquent Object Document Mapper, which allows easy creation, reading, update and deletion of data. The four collections are connected with the core provenance relations as defined by W3C PROV. Each collection is defined by `created_at` and `updated_at` timestamps.

In PROV entities are described with their provenance, that might refer to other entities (i.e. an image is an entity whose provenance refers to other entities such as an annotation on the image, the software component or the agent that created the annotation). Entities can have different attributes and can be described from different perspectives, e.g. a text unit, the same unit after annotation and the aggregation of all annotations on this unit are three distinct entities for which we save provenance. The advantage of using the PROV model inside the CrowdTruth data model is the ability to capture each of the stages performed by the framework (i.e. data pre-processing, gathering human and machine annotations, analyzing the results). Moreover, by capturing all those stages it helps to evaluate the improvement of final results over partial results.

In CrowdTruth **Entities** represent data units and are defined by `format`, e.g. text, image, video with possibility to add other modalities; `domain`, e.g. medical, news, art, also extensible with additional domains; `documentType`, e.g. IBM-medical-sentence, NYT-news-article, Rijks-image; `parents` refers to the parent identifiers to capture the provenance of each data unit, e.g. `wasDerivedFrom` relation and parents are typically generated upon creation of an entity by an activity; `content`, which contains the JSON structure specific to that documentType; `tags`, e.g. unit, segment, frame, which typically can indicate an aggregation level or granularity; `hash` to prevent duplicates in the database; `agent_id` refers to the

agent that `wasAttributedTo` the creation of this entity; `cache`, e.g. batchCount, jobsCount, which is a temporary field for query optimisation.

Agents are defined by a `type`, e.g. user or crowd and are associated with activities and the `softwareComponents_id` used by a specific activity, e.g. `File Uploader` or `CrowdFlower`, i.e. the name of the component. **Activities** refer to the operations performed on entities by a software component or an agent to create a new entity. For example, if the next version of each video, image or text is generated by event annotation, then the activity is this `annotation`. Activities are defined with `used`, `agent_id` and `softwareComponent_id`.

Currently the data model is populated with text, images and videos in three different domains. New data can be ingested in the CrowdTruth MongoDB database through the `Upload Media` option by uploading local files or pulling online resources from APIs. Extending the upload to other domains, types and APIs requires only minimal changes to the framework. Here, we have introduced the main use cases (Section 2), their corresponding annotation tasks (Section 3) and the way the data is stored (Section 4). Next, we describe all CrowdTruth components involved in the end-to-end workflow.

5 The CrowdTruth Framework

The *CrowdTruth* software framework integrates a set of open source components providing an end-to-end workflow for collaborative machine-human computing for annotation of different data modalities (e.g. text, videos, images). To ensure extensibility and openness the framework is implemented using open web standards. It is built on top of an open source PHP framework *Laravel*[7], which uses the MVC pattern to decouple application logic, data and presentation. It leverages built-in packages for authentication, routing, creation of templates and APIs. External packages are used to extend the framework, e.g we use an Object Document Mapper *Moloquent* to query any MongoDB storage. We also developed open source SDKs for CrowdFlower and MTurk to optimise the communication with those platforms. Data ingested and produced through the framework can be exported in different formats. For more details see the documentation[8].

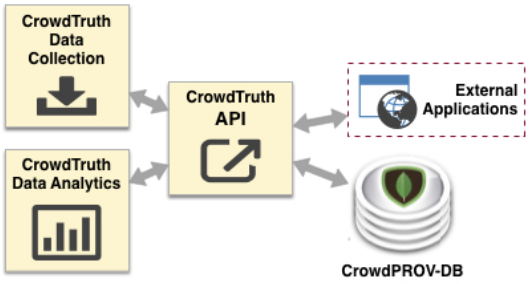

Fig. 3: The CrowdTruth Main Components and Open API

[7] `http://laravel.com/`
[8] `http://crowdtruth.org/info`

Fig. 3 illustrates the framework components. It provides *CrowdTruthPROV-DB*, a provenance-preserving storage of crowdsourcing data, *CrowdTruth Data Collection* services for job configuration, creation and results retrieval, including a library of reusable and extensible micro-task templates, and *CrowdTruth Analytics*, a set of data visualisation and analysis tools. The *CrowdTruth API*[9], is an open API for external applications to query the data in the framework or to ingest their own data. Such an API allows for community building in terms of sharing data, analysis metrics, crowdsourcing templates and optimised job settings. Many of the crowdsourcing templates take a long time to determine their most effective form, thus sharing previous experiences is extremely valuable. Figure 4 provides an overview of the overall framework workflow:

- After input data ingestion, specific *Data Pre-processing* is typically applied to filter out and specify the appropriate input to reach an optimal crowdsourcing task. For examples, the sentence word count property allows for filtering of sentences between a specific word count range.
- The *Job Configuration* component takes the aforementioned filtered input in the form of a batch, and creates a job with specific job settings such as: the crowdsourcing template that is to be used, payment options, and the running platform for the job.
- The *Data Collection* component provides an almost live update of the crowdsourcing results from the annotation platforms, as these results are pushed from CrowdFlower and polled at regular intervals from MTurk. The results are stored in the database along with their provenance.
- *Post-processing* allows for deep analysis of the quality of the crowdsourcing results on three levels: Worker, Annotation and Unit. The *CrowdTruth Metrics* are able to identify the low quality workers, the suitability of a unit for a task and the clarity of the annotations.
- The *Data Analytics* component provides visualizations tailored for use with the CrowdTruth metrics. As such, it provides functionalities for evaluating results through graphical views at both individual and aggregated levels.

The following sub-sections describe each component in more detail.

5.1 Data Pre-processing Components

The pre-processing components allow for various processing of the input data to optimize its use in specific crowdsourcing tasks. Before running a *flower name annotation task* we pre-process images to know which ones have high probability of depicting a flower and we send only those for crowd annotations. This saves both cost and time and makes the micro-task more engaging for the workers. Figure 5 depicts the three **pre-processing workflows** for all content modalities. The left side (A) of the figure shows the workflow for **video and image pre-processing** and the right side (B) shows the workflow for **text pre-processing**. They all share the same MongoDB storage (depicted in the centre of the figure). The video pre-processing makes also use of a physical storage. Following, we provide details on the three pre-processing workflows in this figure.

[9] `http://crowdtruth.org/api/examples`

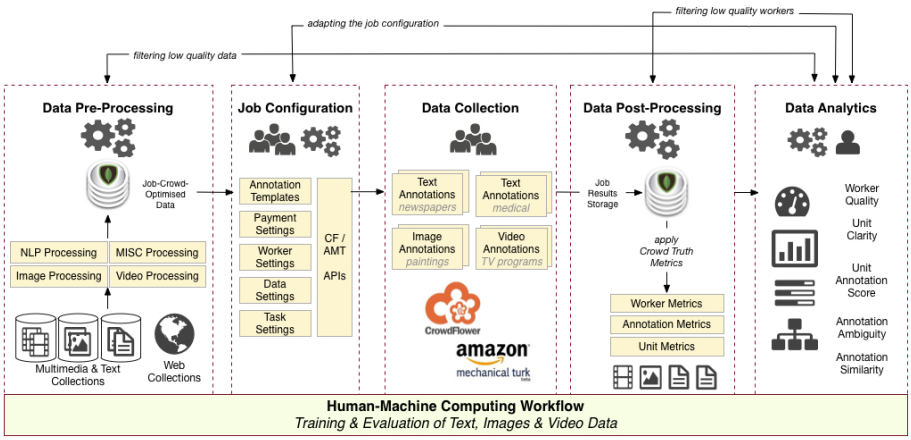

Fig. 4: CrowdTruth Overall Architecture

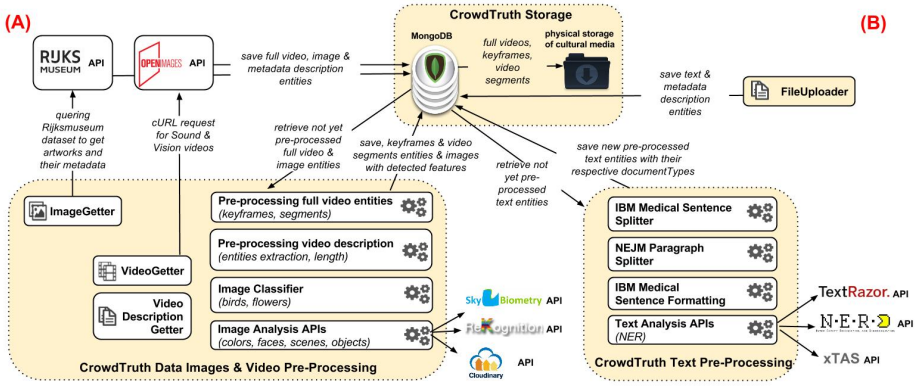

Fig. 5: CrowdTruth Pre-processing Workflows for Text, Images and Videos

To **ingest images in CrowdTruth framework** we use `ImageGetter`, which calls the open API of the Rijksmuseum Amsterdam[10] by querying, e.g. for a number of paintings or drawings described with a specific keyword, like 'birds'. It is straightforward to extend it with additional APIs of other online collections. The **Image pre-processing** is performed by three external APIs - Rekognition[11], Cloudinary[12], Skybiometry[13], and a local classifier. Each of them contributes complimentary and redundant annotations with their corresponding confidences, e.g. Rekognition provides depicted objects, faces; Cloudinary detects faces, colour histogram, while Skybiometry detects faces with their position and gender. The local classifier is trained for flowers and birds. The pre-processing

[10] `http://www.rijksmuseum.nl/api`
[11] `http://rekognition.com/`
[12] `http://cloudinary.com/`
[13] `http://www.skybiometry.com/`

is finalised by storing the image URLs and metadata in the MongoDB database as parent entities together with separate children. The children entities contain information about the software agent used and its configuration, as well as the features received by calling the aforementioned APIs and the classifier.

To **ingest videos in CrowdTruth framework** we use OpenImages[14] API by querying for videos from the collection of the Netherlands Institute for Sound and Vision. Figure 5 on the left (A) depicts the workflow for *video pre-processing*. After returning the requested number of videos from OpenImages, we create an entity for each item, containing all the metadata features. The item is linked through the provenance model to an activity `OpenImagesGetter` and an agent, e.g. CrowdTruth user. Next, each video is downloaded and saved in the public storage of the framework together with its description as Metadata Description entity. For maintaining the provenance consistency, the Metadata Description entities are linked to an activity `VideoDescriptionGetter`, an user_id and the full video as the parent entity.

To optimise the crowd annotations, videos need to be pre-processed to a length reasonable for a micro-task, e.g. up to a minute. Thus, we perform video segmentation. Similarly as with the images, we would like to have some indication of the featured topics and objects in each video. For this we extract keyframes, which are processed as images to detect the depicted objects. Both pre-processing are implemented using the open source FFmpeg[15] framework. Additionally, to detect main concepts we process the video description and transcript and extract the named entities. The new entities get stored in the database with their particular activity, user and parent entity.

We **ingest text in CrowdTruth framework** using a local component `FileUploader`, as we are provided with large amounts of IBM Watson medical data to experiment with. The text pre-processing is depicted in the right part (B) of Figure 5. Text annotation tasks typically require specific formatting of the text in order to anchor the human annotation around specific word(s) or phrase(s). Similarly as with the videos, the text needs to be fitted to a length suitable for a micro-task, e.g. sentences or short paragraphs. Additional filters to maximise the quality of the sentences have also been implemented, e.g. detection of UMLS[16] medical relations, semicolon or comma-separated list in sentences. For detailed examples of those **special filters** consult the dedicated document section `http://crowdtruth.org/info/special_filters`.

Additionally, for the *Event extraction from newspapers* task, we have ingested a set of NYTimes article URLs and extracted the date when the article was published and its content. Pre-processing activities for these texts are (1) sentence splitting, (2) length-based selection on the sentences for removing too short sentences which are meaningless, (3) putative events extraction using the Stanford Parser[17] (mainly for verbs) and NomLex, a dictionary for nominaliza-

[14] `http://www.openbeelden.nl/api/`

[15] `http://www.ffmpeg.org/`

[16] `https://uts.nlm.nih.gov/home.html`

[17] `http://nlp.stanford.edu/software/lex-parser.shtml`

tions. Next, (4) the putative event is marked in the sentence with capital letters and surrounded by square brackets; and (5), for each event or event role filler (participants, location, time) we align their types to a set of predefined (existing but simplified) ontologies (Table 1).

5.2 Job Configuration and Data Collection Components

The Job Configuration component provides functionality for (1) creation of batches of media units to be used in a job, (2) job template configuration and (3) job settings (Fig. 6). Each job can be duplicated or adapted for different data, settings and template which is saved in a JSON format and further translated to the dedicated crowdsourcing platform format. The platform components are written in the form of Laravel packages. In the documentation there is information on how to write your own package, by extending an abstract class, calling your API and adhering to our data model standard. After configuring the job's title, reward and other settings, the user creates the job. The request is routed through the respective package, where any necessary conversion is done, to the platforms' API. If this succeeds, one job per platform is stored in our database.

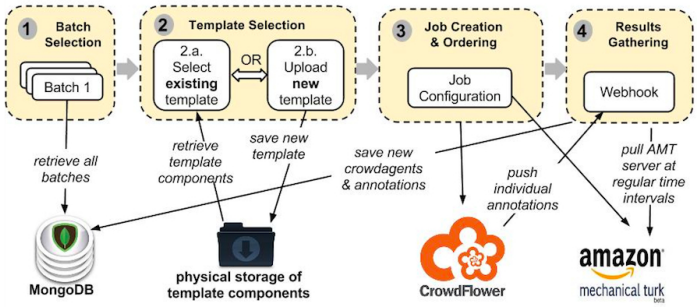

Fig. 6: CrowdTruth Job Configuration and Data Collection Workflow

The **Collection of Annotated Data** in CrowdTruth is a workflow of four main steps as depicted in Figure 6. It starts with the steps described for the **Job Configuration** component: batch creation, template selection and job creation and ordering. Finally, in the *Results Gathering* phase the crowdsourcing results from both CrowdFlower (webhook call when a new judgement is received) and MTurk (poll the mTurk server at regular intervals to check for new judgements) are pulled into CrowdTruth framework. Results are saved in the MongoDB database in the PROV model, along with each additional information provided by the platforms.

5.3 Data Post-Processing and Data Analytics Components

Data visualization plays a central role in the CrowdTruth framework. It provides tools for deep analysis of crowd data based on the core notion of CrowdTruth, to harness disagreement. Ultimately, it should implement the instantiation of the triangle of reference [3] for the range of tasks supported in the framework. The *Data Analytics* component is developed using the Highcharts JS library and interacts with the CrowdTruth API. In the backend the requests are processed into

optimized aggregated queries for the MongoDB database. Thus, the data is protected and the process is optimized by efficiently querying the DB and partially executing in the backend the necessary computations. On one hand the interface is more responsive, increasing the framework usability. On the other hand, the visual components are synchronized and communicate between themselves, e.g. general and specific information views, as well as their table views.

The visual components depict the three main sections of the framework: media, workers and jobs. The views facilitate the visualization and analysis of imported and generated data by the framework (media, workers, jobs). The visualization of new data is possible as long as it conforms to the defined data model. All the charts are created through a facade object which specifies the settings of the graphs. Thus, the charts are easily adaptable by changing the settings of the objects to be created. Beside the barchart views, which are specific to each section, all the other components of the views share the same implementation making the framework robust to changes and easily extensible.

The core of the CrowdTruth framework are the disagreement metrics [6, 5] that evaluate evaluate the crowdsourced data in a variety of annotation settings, such as event extraction, video and image annotation, medical relation and factor extraction. These metrics are implemented in Python and similarly to the visualization component use the API to get the data from the server. The basic assumption of the framework and metrics is that each individual unit that can be interpreted (e.g. sentence, image, video) is annotated by multiple workers, and their annotations are aggregated together and used in the following ways:

Annotation vector: The most important step in adapting the CrowdTruth metrics to a new task is designing the annotation vector so that the results can be compared using cosine similarity. For each worker i submitting their solution to a micro-task on a MediaUnit u, the vector $W_{u,i}$ records their answers. If the worker selects an answer, its corresponding component would be marked with '1', and '0' otherwise. The size of the vector depends on the number of possible answers per task. The output for open-ended tasks (e.g. FactSpan) was interpreted to fit into a fixed-size vector, for the purpose of reusing the disagreement metrics. An explanation of how the Annotation vectors were adapted for various crowdsourcing tasks is available in Table 2.

MediaUnit vector: This vector accounts for all worker submissions on a unit, for a given task. For every unit u, we compute the MediaUnit vector $V_u = \sum_i W_{u,i}$ by adding up the annotation vectors for all workers on the given task. Along with the Annotation vector, this is used as a core component for analysing disagreement in the crowd.

The crowdsourcing system contains 3 components: the Worker, the Unit, and the Annotation. Ambiguity can occur as part of each of these components (e.g. a spammer can generate disagreement for the Worker component) or can propagate inside the system (e.g. an unclear Unit can generate disagreement among workers). Therefore, we analyse how ambiguity and disagreement occur for each system component using the Annotation and MediaUnit vectors and a set of specialized metrics for Worker, Annotation and Unit. We use the cosine

Table 2: Annotation vectors for the various crowdsourcing tasks

Task	**Annotation vector**
FactSpan	9-component vector: 3-words-left-of-factor, 2-words-left-of-factor, 1-words-left-of-factor, factor, 1-words-right-of-factor, 2-words-right-of-factor, 3-words-right-of-factor, OTHER, Answer-Validation
RelEx	16-component vector: 12 components - each corresponding to a relation including NONE and OTHER, and Answer-Validation-NONE, Answer-Validation-OTHER
RelDir	3-component vector: each possible direction of a relation and no relation
RelExDir	23-component vector: each relation with its inverse (if it exists)
EventEx	14-component vector: each event type, OTHER and NONE
PartEx, LocEx, TimeEx	the size of the vector corresponds to the number of defined types for Location, Time, Participants + OTHER and NONE
Passage filtering	2 annotation vectors: one to account for disjoint passage-answer pairs, and one for multiple-choice justifications
Passage alignment	fixed-size vectors for each question-passage pair, with a component for each type of relation that can exist between the terms

similarity coefficient as the basis of most of these metrics, in order to determine the similarity of vectors. In the following section, we show both their definition and examples of visualisation in the CrowdTruth Analytics (see Fig. 7, 9, 10).

5.4 Worker Metrics

These metrics are used to measure disagreement at the level of the worker, in order to differentiate between spammers and high quality contributors.

Worker-unit disagreement measures the cosine distance between a worker's Annotation vector and the MediaUnit vector (subtracting the worker vector), for each Worker-Unit pair. The average of this metric across all units in a set gives a measure of how much a worker disagrees with the crowd on a per-unit basis. Consistent low unit disagreement scores can indicate a low quality worker.

Worker-worker disagreement is equal to $1 - avg(\kappa)$ for a particular worker. Since κ is a pairwise metric, for each worker we average the κ scores between that worker and all the others. Similarly to the previous metric, the worker-worker disagreement metric measures how close a worker performs to the group of workers solving the same tasks. sagreement with the majority of workers is an indicator of low quality work.

Average annotations per unit is measured for each worker as the number of annotations they choose per unit averaged over all the units they annotate. Since in many tasks workers are allowed to choose "all annotations that apply", a low quality worker can appear to agree more with the crowd by repeatedly choosing multiple annotations, thus increasing the chance of overlap. A high score here can indicate low quality workers. All three metrics are used to determine worker quality in the pie chart on the left in Fig. 7.

5.5 Unit Metrics

These metrics are used to determine the clarity of the input unit that is given to the crowd. An ambiguous unit (e.g. a sentence that is difficult to read) could generate disagreement, therefore tampering with the quality of the results.

Unit-annotation score is the core CrowdTruth metric. It is measured for each annotation on each unit as the cosine similarity of the unit vector for the annotation with the MediaUnit vector. For instance, in Fig. 8, unit 735 has complete

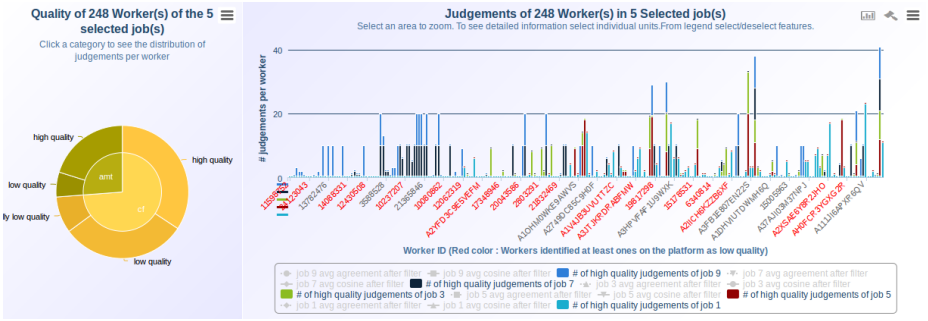

Fig. 7: Screenshot of CrowdTruth Analytics for Worker Quality and Annotations (jobs comparison); more details can be obtained by clicking on a worker (bar chart), or a type of worker (pie chart)

Rel: 15 Workers/sent pair														Sentence Clarity
Sentence ID	sT	sP	sD	sCA	sL	sS	sM	sCI	sAW	sSE	sIA	sPO	sNONE	sOTH	
225527731	0	0	0	1	0	11	0	0	0	0	0	0	0	0	0.91
225527732	0	0	0	0	0	7	2	0	2	2	0	1	0	0	0.50
225527733	0	0	0	1	0	7	1	0	1	0	0	0	0	1	0.63
225527734	0	0	0	0	0	1	0	0	2	0	0	0	0	9	0.75
225527735	0	0	0	0	0	13	0	0	0	0	0	0	0	0	1.00

Fig. 8: Annotation vectors of RelEx task on 5 units, with 15 workers contributing per unit. Rows are individual units, columns are the annotations. Cells contain the number of workers that selected the annotation for the unit, i.e. 7 workers selected the *sS* annotation for unit 732. The cells are heat-mapped per row, highlighting the most popular annotation(s) per unit.

agreement between annotators for annotation *sS*. Therefore, the unit-annotation score for unit 735 and annotation *sS* is equal to 1. Unit 733 has more disagreement, so its unit-annotation score for annotation *sS* is equal to 0.63.

Unit clarity is defined for each unit as the maximum annotation score for that unit. This metric is used to determine the quality of the unit which is given as input to the crowd. If all the workers selected the same annotation for a unit (e.g. unit 735 in Fig. 8), the max annotation score will be 1, indicating a clear unit. In contrast, unclear units will have low clarity scores (e.g. unit 732 has a clarity score of 0.5). Unit clarity is shown in Fig. 9, among other worker and annotation metrics. This view is the most comprehensive tool to compare sub-sets of MediaUnits (containing one ore more units) with each other.

5.6 Annotation Metrics

These metrics are used to measure the quality of the pre-defined annotation types that are part of the task (e.g. whether or not relations in *RelEx* have overlapping meanings). This can then be used to distinguish between disagreement that is the result of low quality workers, and the disagreement from badly designed tasks, in order to improve future crowdsourcing.

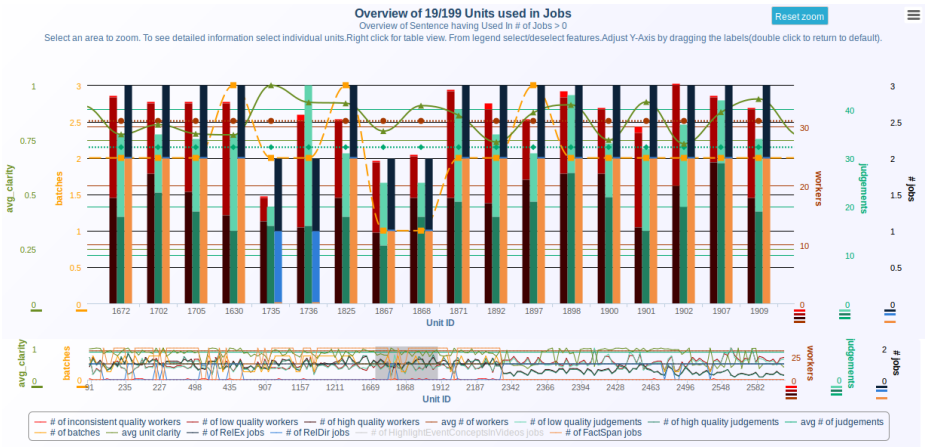

Fig. 9: Screenshot of CrowdTruth Analytics for Units; more details about the unit jobs, workers and annotations can be obtained by clicking on a unit bar

Annotation similarity is defined as the *causal power* [13], which is the pairwise conditional probability $P(A_j|A_i)$ adjusted for the prior probability of A_i. We want to know if annotation A_i is annotated in a unit and how often annotation A_j is as well, but only if A_j is significantly more likely to be annotated when A_i is as well. A high similarity score for a pair of annotations indicates the annotations are confusable to workers: their semantics may be similar or routinely expressed in similar ways in language, or the semantic specification may be confusing or vague. For example when annotations for two relations often appear together in sentences, this could mean the relations are confusing, overlapping in meaning, etc. In Fig. 8, the sCA and sS annotations appear to have this form of similarity.

Annotation ambiguity is defined for each annotation as the maximum annotation similarity for the annotation. If an annotation is clear, its score is low. Annotation that is strongly associated with other may create problems for the task, as well as for training machines that need to discern between them.

Annotation clarity is defined for each annotation as the max unit-annotation score for the annotation over all units (of a given type). If an annotation has a low clarity score this may indicate unattainable NLP targets and problems with the semantic specification. For instance, in Fig. 8, sM is one example of a low-clarity annotation, since few workers ever picking this annotation.

Annotation frequency is the number of times the annotation is annotated at least once in a MediaUnit. The latter three metrics are shown in Fig. 10.

6 Related Work

The amount of knowledge that crowdsourcing platforms like CrowdFlower or Amazon Mechanical Turk hold fostered a great advancement in human computation [14]. Although the existing paid platforms manage to ease the human computation, it has been argued that their utility as a general-purpose computation platform still needs improvement [9]. Both paid platforms support the task creation, distribution to the workers and gathering of the results and provide

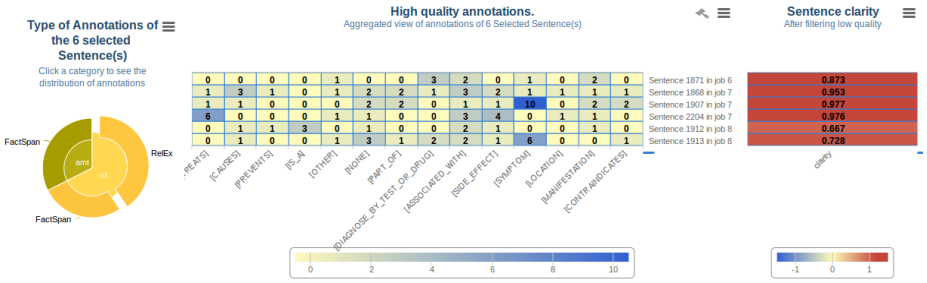

Fig. 10: Screenshot of CrowdTruth Analytics for Annotations on Selected Units in Selected Jobs; click on the pie chart to see the annotation distribution per micro-task

some quality management tools. However, the quality measures that they apply are inferior to our CrowdTruth metrics, as lots of tainted judgements are still accepted. Even if CrowdFlower's job monitoring support improves the analysis of the data, the provided set of quality metrics is limited. Moreover, the missing links for interconnecting the units, workers and annotations across one or multiple jobs hinder the data exploration and visualization.

Since the development of crowdsourcing has become more intensive, much research has been done in combining human and machine capabilities in order to obtain an automation of the process. Some state-of-the-art crowdsourcing frameworks are CrowdLang [9], CrowdMap [15], GATECrowd [11]. CrowdLang represents a general approach of integrating both human and automatic computation for different use cases and media modalities. However, it restricts the users to work with its own internal programming language, while the overall framework availability for usage or testing is still low. Further, CrowdMap represents an implementation of a workflow model for crowdsourcing mappings between ontologies. The main drawback of the framework is the fact that its parameters are tuned to get the best results for ontology alignment tasks, and it is not easily extendable to other types of media formats or tasks. Furthermore, both frameworks lack in proper visualization of the annotated data.

A more general solution for language processing is represented by GATECrowd, a crowdsourcing plugin for the GATE framework. It facilitates the preprocessing of data for crowdsourcing tasks, communicates with Crowdflower for gathering the annotated data and aggregates the results. The plugin takes advantage of the GATE toolbox functionalities for collecting and processing the data, calculating the inter-annotator agreement and analysis of the data. Additionally, the quality of the results is insured through golden units. Similarly to CrowdMap, one of the disadvantages of GATE is its limitation to text media types. By capturing the provenance between the machine and human generated annotations, the creation of new metrics is possible. However, additional metrics imply existing implementation inside the GATE architecture, which introduces the overhead of familiarization with the entire GATE architecture.

A lot of research has been focused on identifying crowdsourced spam. Although a commonly used algorithm for removing spam workers is the majority decision [16], according to [17] it is not an optimal approach as it assumes all

the workers to be equally good. Alternatively, expectation maximization [18] estimates individual error rates of workers. First, it infers the correct answer for each unit and then compares each worker answer to the one inferred to be correct. However, [6] shows that some tasks can have multiple good answers, while most spam or low quality workers typically select multiple answers. For this type of problem, some disagreement metrics [5] have been developed, based on workers annotations (e.g. agreement on the same unit, agreement over all the units) and their behavior (e.g. repetitive answers, number of annotations).

7 Conclusions and Future work

In this paper, we introduced the CrowdTruth open-source software framework as an end-to-end collaborative machine-human computing workflow for text, images and video annotations across different domains and use cases. *CrowdTruth framework* implements the novel *CrowdTruth Methodology* for gathering annotated data, which rejects the notion that human interpretation can have a single *ground truth*, and is instead based on the observation that disagreement between annotators can signal ambiguity of the content or annotation task. The CrowdTruth methodology is based on the *triangle of reference* [3] whose implementation in the framework allows for easy adaptation to new micro-tasks. We have validated this, as the initial set of metrics was developed for the medical text use case of IBM Watson and we easily applied them to new tasks, such as event extraction in newspaper text, question-answer alignment and video and image annotations.

We presented the details of the entire human-computing and machine processing workflow, as well as the specifics of each framework component. We demonstrated how such a framework can be beneficial to the Semantic Web community by adding human semantics to existing content interpretations, as well as by supporting the growing trend for crowdsourcing tasks, and continuous need for gold standard data. Detailed documentation `http://crowdtruth.org/info` and code `https://github.com/laroyo/CrowdTruth` are provided online. Data export from the CrowdTruth framework is provided in different formats and at different phases of the workflow. The CrowdTruth framework is implemented using open standards, and an important gain is achieved by the usage of the PROV model, compared to existing crowdsourcing platforms and frameworks. This ensures a monotonically increasing behaviour curve in terms of media unit clarity and micro-task template suitability for each media unit that is intended to gather annotations. As future work, we plan to gather more use cases to extend the system with new data, micro-task templates and domains. Additional visualisations are also explored to increase the usability and effectiveness of the CrowdTruth metrics.

References

1. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: Proceedings of CEMNLP'08, Association for Computational Linguistics (2008) 254–263

2. Ambati, V., Vogel, S., Carbonell, J.G.: Active learning and crowd-sourcing for machine translation. In: LREC. Volume 1., Citeseer (2010) 2
3. Aroyo, L., Welty, C.: Truth is a lie: Crowdtruth and the 7 myths of human annotation. AI Magazine (2014)
4. Aroyo, L., Welty, C.: Crowd Truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. WebSci2013 (2013)
5. Aroyo, L., Welty, C.: Measuring crowd truth for medical relation extraction. In: AAAI2013 Fall Symp. on Semantics for Big Data. (2013)
6. Soberón, G., Aroyo, L., Welty, C., Inel, O., Lin, H., Overmeen, M.: Measuring crowd truth: Disagreement metrics combined with worker behavior filters. In: Proc. of CrowdSem2013 Workshop), ISWC2013. (2013)
7. Inel, O., Aroyo, L., Welty, C., Sips, R.J.: Exploiting Crowdsourcing Disagreement with Various Domain-Independent Quality Measures. In: Proc. of DeRiVE2013 Workshop), ISWC2013. (2013)
8. Aroyo, L., Welty, C.: Harnessing disagreement for event semantics. In: Proc. of DeRiVE2012, ISWC2012. (2012) 31
9. Minder, P., Bernstein, A.: Crowdlang-first steps towards programmable human computers for general computation. In: Human Computation. (2011)
10. Ahmad, S., Battle, A., Malkani, Z., Kamvar, S.: The jabberwocky programming environment for structured social computing. In: Proceedings of ACM symposium on UI software and technology, ACM (2011) 53–64
11. Bontcheva, K., Roberts, I., Derczynski, L., Rout, D.: The gate crowdsourcing plugin: Crowdsourcing annotated corpora made easy. In: Proc. EACL. (2014)
12. Groth, P., Moreau (eds.), L.: PROV-Overview. An Overview of the PROV Family of Documents. W3C Working Group Note NOTE-prov-overview-20130430, World Wide Web Consortium (April 2013)
13. Cheng, P.: From covariation to causation: A causal power theory. Psychological Review (104) (1997)
14. Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. In: Proc. of the SIGCHI, ACM (2011) 1403–1412
15. Sarasua, C., Simperl, E., Noy, N.F.: Crowdmap: Crowdsourcing ontology alignment with microtasks. In: The Semantic Web–ISWC 2012. Springer (2012) 525–541
16. Hirth, M., Hoßfeld, T., Tran-Gia, P.: Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In: Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE (2011) 316–321
17. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. The Journal of Machine Learning Research **99** (2010)
18. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the em algorithm. Applied Statistics (1979) 20–28