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 in a wide range of annotation tasks and on a variety of media (e.g. text, images, videos). The CrowdTruth approach captures human semantics through a pipeline of three processes: *a*) combining various machine processing of text, image and video in order to understand better the input content and optimise its suitability for micro-tasks, thus optimise 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; and *c*) implementing 'disagreement metrics', i.e. *CrowdTruth metrics*, to support deep analysis of the quality and semantics of the crowdsourcing data. Instead of the traditional inter-annotator agreement, we use their disagreement as a useful signal to evaluate the data quality, ambiguity, and vagueness. In this paper we demonstrate the innovative CrowdTruth approaches embodied in the software to: *1*) support processing of different text, image and video data; *2*) support a variety of annotation tasks; *3*) harness worker disagreement with CrowdTruth metrics; and *4*) provide an interface to support data analysis and visualisation. In previous work we introduced the CrowdTruth methodology with examples for semantic interpretation of medical text for relation and factor extraction, and with newspaper text for event extraction. In this paper, we demonstrate the applicability and robustness of the approach to a wide variety of problems across a number of domains. We also show the advantages of using open standards and the extensibility of the framework with new data modalities and annotation tasks, as well as its openness to external services.

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] for gathering human interpretation on data has become now a mainstream process. In AI this has become a scalable way to gather a cheaper annotated data for gold standards that are used to train and evaluate machine learning systems. In NLP, crowdsourcing has been used for nearly a decade, as the low level language understanding tasks map well into crowdsourcing micro-tasks. However, as we have observed previously [1], the introduction of crowdsourcing has not fundamentally changed the way gold standards are created; in particular, 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 [2] and by the observation that disagreement is a natural product of having multiple people perform annotation tasks, and as such can provide useful information about the task, a particular annotation unit, 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 [3]. In [2, 4–6] we have validated *CrowdTruth* in the context of measuring the quality of workers, annotation units, and tasks. We showed experimental evidences that these measures are inter-dependent, and that existing crowdsourcing approaches that measure only worker quality are missing important information, as not all sentences are created equal.

In this paper, we present 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

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

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

for gold standard training and evaluation data. The open source CrowdTruth framework is available for download at `https://github.com/laroyo/CrowdTruth` and the service at `http://crowdtruth.org`.

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 diversity in the data, each use case introduces either a new domain, content modality or a new annotation task. All the data from these use cases and experiments can be viewed in *CrowdTruth* through the *Media* section. New content can be uploaded for each of those use cases to run new experiments anytime. CrowdTruth provides *Upload Media* option, as described in more details in Section 4. Below we describe the five use cases:

- IBM Watson *medical text* annotation for *relation extraction* (RelEx)
- IBM Watson *medical text* annotation for *factor span extraction* (FactSpan)
- 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)

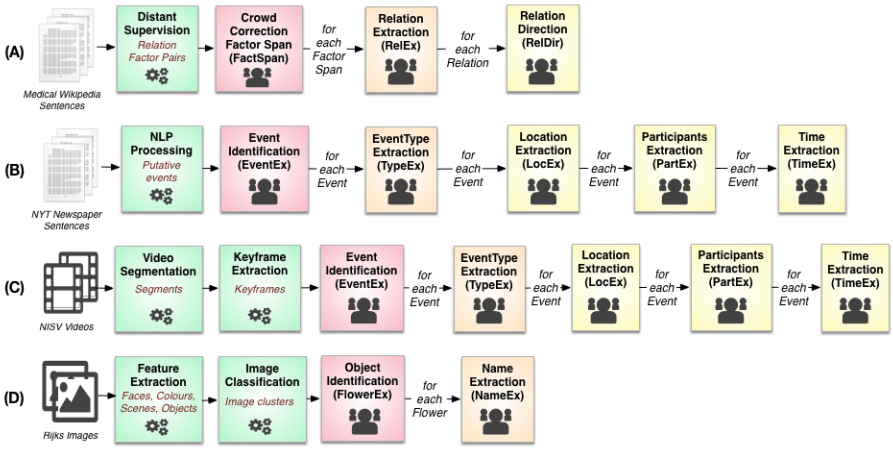

Fig. 1: CrowdTruth Annotation Workflows for Text, Images and Videos

The best illustration on how the CrowdTruth Framework works can be observed currently in the data on the *RelEx* and *FactSpan* use cases. The reason for this is, that 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. 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 entity 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 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 (e.g. text, images, videos) and three domains (e.g. 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 extendable *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: Correction Factor Span**. The crowd is given a *sentence* with two highlighted $factors$. 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 Type Identification**. The crowd is given a *sentence* with two highlighted $factors$ and a set of 12 target $relationtypes$. The crowd is asked to select all relation types that are 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 to keep the workers honest.
- **RelExDir: Relation Extraction & Direction Identification**. The crowd is given the combined task of relation extraction and direction on the output from $FactSpan$. As with $RelEx$, the workers are shown a sentences with the two highlighted $factors$ from the $FactSpan$ task, and then are asked to check all relations that apply between them. On each selected relation its direction is also asked.

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 *putativeevent* 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, Region, City, State, Area on Land - Valley, Island, Mountain, Beach, Forest, Park, Area on Water - Ocean, River, Lake, Sea, Road/Railroad - Road, Street, Railroad, Tunnel, Building - Educational, Government, Residence, Commercial, Industrial, Military, Religious
Period	Before, During, After, Repetitive, Timestamp, Date, Century, Year, Week, Day, Part of Day
Participants	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**. In the pre-processing named entity are extracted from 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*, i.e. literally mentioned in the video, or *associated*, i.e. related to some spoken events/role fillers in the video.

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 types of metadata and content modalities such as text, images, videos, (2) to be specific enough, i.e. semi-structured, to still be able to query this data, and (3) to capture the provenance for the data stored. The MongoDB 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 annotation data, we often do not know upfront what structure will be appropriate. An example of this are the various online content processing APIs that return results in a specific JSON format with different structure for every API. 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).

The CrowdTruth MongoDB deployment hosts one database, with four collections `Entities`, `Activities`, `Agents` and `SoftwareComponents`. For every

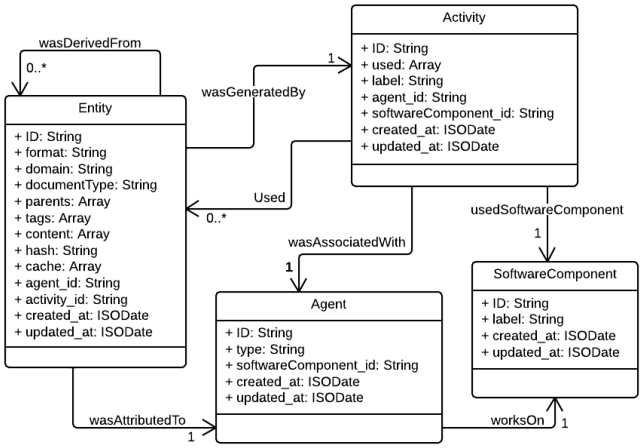

Fig. 2: The CrowdTruth Data Mode and Data Provenance

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, updating and deletion of data. The four collections are connected with the core provenance relations as defined by W3C PROV [6]. 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. For example, an image is an entity whose provenance refers to other entities, such as, an annotation on the image, and the software component and the agent for the creation of that annotation. Entities can have different attributes and can be described from different perspectives, e.g. a text unit in CrowdTruth, the same unit after annotation, and the aggregation of all annotations on this unit are three distinct entities for which we describe provenance. In CrowdTruth **Entities** represent the data units and are defined by `format`, e.g. text, image, video with possibility to extend with additional modalities if needed; `domain`, e.g. medical, news, art, also extensible with additional domains; `documentType`, e.g. IBM-medical-sentence, NYT-news-article, NISV-video, Rijks-image; `parents` provides reference to the parent identifiers to capture the provenance of each data unit, e.g. based on `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, etc, 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, etc, 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. `FileUploader` or `CrowdFlower`, i.e. the name of the component. **Activities** re-

[6] http://www.w3.org/TR/prov-primer/#intuitive-overview-of-prov

fer 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 uploads to other domains, types and APIs require 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 into separate components. It leverages the built-in packages for authentication, routing, creation of templates and APIs. Additional external packages are used to extend the framework. For example, we use an Object Document Mapper *Moloquent* to query any MongoDB data storage. We also developed open source SDKs for CrowdFlower and Amazon Mechanical Turk 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].

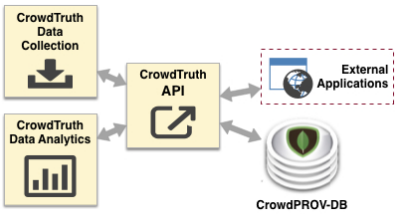

Fig. 3: The CrowdTruth Main Components and Open API

Fig. 3 illustrates framework components. It provides *CrowdTruthPROV-DB*, a provenance-preserving storage of crowdsourcing data, *CrowdTruth Data Collection* services for job configuration, creation and result retrieval, including a library of reusable and extendable micro-task templates, and *CrowdTruth Analytics*, a set of data visualisation and analysis tools, for a deep analysis of crowdsourcing data. The *CrowdTruth API* [9], is an open API for external appli-

[7] http://laravel.com/
[8] http://crowdtruth.org/info
[9] http://crowdtruth.org/api/examples

cations 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, which is described in the following sub-sections.

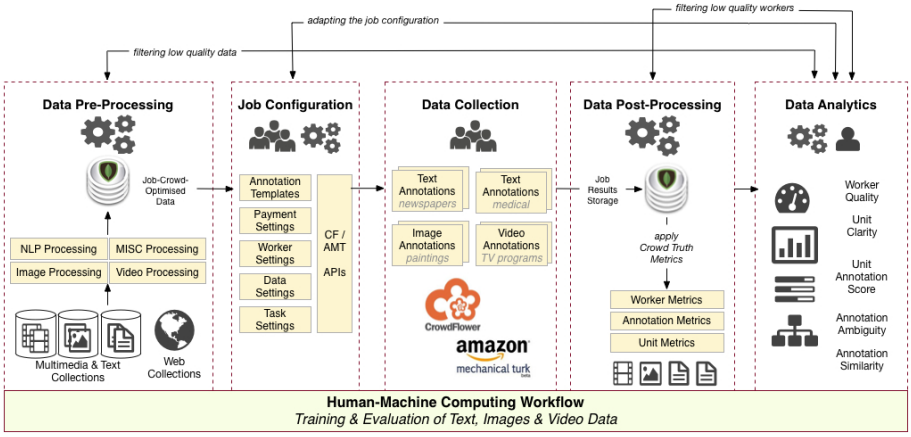

Fig. 4: CrowdTruth Overall Architecture

5.1 Data Pre-processing Components

The pre-processing components allow for a various type of processing of the input data to optimise its use in specific crowdsourcing tasks. For example, before running a *flower name annotation task* we pre-process images to know which ones have high probability of depicting of a flower and we send only these 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 additionally makes use of a physical storage for the full videos. Following, we provide details on the three pre-processing workflows in this figure.

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 Rijksmuseum API implements oai_dc (Dublin Core) metadata format and Europeana Data Model, which is currently the open standard for online cultural heritage collections. The **Image pre-processing** is performed by

[10] http://www.rijksmuseum.nl/api

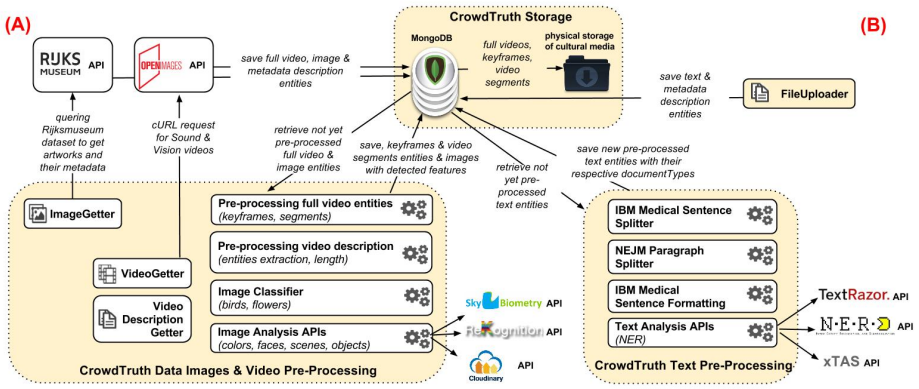

Fig. 5: CrowdTruth Pre-processing Workflows for Text, Images and Videos

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 object, scene and face detection; Cloudinary detects number of faces, main colours and colour histogram, and Skybiometry detects faces with their position and gender. The local classifier is trained for the domains of flowers and birds. The performance is evaluated using four-fold cross validation based on modified F1-score punishing stronger for wrong classification above a threshold higher than 0.75. The pre-processing is finalised by storing the image URLs and metadata in the MongoDB database as parent entities together with separate children referring to the proper parent. The children entities contain information about used software agent and its configuration, as well as the respective features received by calling all aforementioned APIs and the classifier. Additional training for other concepts (next to flowers and birds) can be easily provided as an extension of this component.

To **ingest videos in CrowdTruth framework** we use OpenImages[14] open API by querying for videos (URL and metadata) from the collection of the Netherlands Institute for Sound and Vision. Figure 5 on the left (A) depicts the workflow for **video pre-processing**. The media collection pipeline is performed by means of successive API calls. Each request is written in PHP and uses the cURL library. After returning the requested number of videos from the OpenImages platform, we create a parent entity for each item, that contains all the features provided by the metadata and 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 the abstract component of the metadata as metadata description entity. For maintaining the provenance consistency, the metadata description entities

[11] http://rekognition.com/
[12] http://cloudinary.com/
[13] http://www.skybiometry.com/
[14] http://www.openbeelden.nl/api/

are linked to an activity `VideoDescriptionGetter`, an user_id and to 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] multimedia framework. For key frames extraction we use the scene detection filter implemented by FFmpeg which uses values between 0 and 1 to indicate a new scene, thus the lower the value, the lower the probability of introducing a new scene of the video. To keep a balance between the length of the video and the number of keyframes, we set this value to 0.5. Additionally, to detect main concepts we process the video description and transcript. The new entities get stored in the database with their particular activity `KeyframeExtraction`, user and parent entity.

We **ingest text in CrowdTruth framework** with the help of a local component `FileUploader`, as we were 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. To realise this we have developed a `SentenceSplitter1` and `ParagraphSplitter` (currently adapted to the specific format of the IBM Watson medical annotation task). Additional filters to maximise the quality of the sentences have also been implemented, e.g. detection of UMLS [16] medical relations in sentence, detecting semicolon or comma-separated list in sentences, etc. For detailed examples of those **special filters** consult the dedicated document section `http://crowdtruth.org/info/special_filters`.

Additionally, for the "Event extraction from newspaper text" task, we have ingested a set of NYTimes article URLs and applied HTML DOM parser for extracting the date when the article was published and the entire content of the article. Pre-processing activities for these texts are (1) `SentenceSplitter2` using the `DocumentPreprocessor` from Stanford Lexical parser[17], (2) length-based selection on the sentences for removing too short sentences, which are meaningless, (3) putative events extraction, i.e. all the verbs, mostly indicating the predicate of the sentence and all the nominalized verbs, mostly indicated through nouns. Two main classes of Stanford parser are used: `LexicalizedParser` set to English, which creates the grammar representation of each sentence and `Typed-Dependency`, which creates relationships between grammatical instances within the sentence. NomLex (dictionary of nominalizations) is used to get the nominalized verbs; (4) the putative event in marked in the sentence with capital

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

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

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

letters and surrounded by square brackets; and (5) for each event role filler we provide ranges to align events, their types and corresponding participants, location and time to a set of predefined (existing but simplified) ontologies (Table 1).

5.2 Job Configuration Components

The Job Configuration component provides functionality for the (1) creation of batch of media units to be used in a job, (2) job template configuration and (3) job settings. Each job can be duplicated or adapted for different data, settings and template. Amazon Mechanical Turk and CrowdFlower both allow additional CSS and JS to be used, which are also stored in the framework. All this is saved in a JSON format and further translated to the dedicated crowdsourcing platform format. In this way, we can ensure that the setting of the job is suited to measure its results with the CrowdTruth metrics. The platform components are written in the form of Laravel packages. New platforms are added by registering them in the configuration files. 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. We included our own SDK's for communicating with mTurk and CrowdFlower. The list of possible platforms is found by the framework by checking in the configuration files which platforms are installed. After configuring the job's title, reward and other settings, the user finally creates the job. The request is routed through the respective package, where any necessary conversion is done, to the platforms' API. If this succeeded, one job per platform is stored in our database.

5.3 Data Collection Components

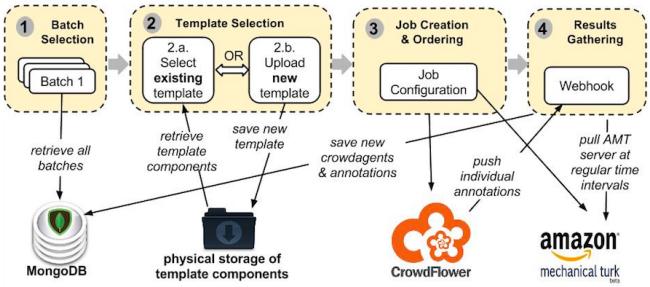

Fig. 6: CrowdTruth Data Collection Workflow

The **collection of annotation data** in CrowdTruth is a workflow of four main steps as depicted in Figure 6. It starts with the *Batch Creation*, where the target content units are selected to be used in a new job. The *Task Template Selection* then helps with the assigning of the appropriate task template for the chosen content units in step 1. The *Job Creation & Ordering* further allows to select the suitable job configuration. Finally, in the *Results Gathering* phase the crowdsourcing results from both CrowdFlower (webhook call when new a judgement is received) and AMT (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 Open Provenance Model, along with all additional information each of the platforms provides.

5.4 Data Analytics Components

Data visualisation 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 annotator disagreement and ultimately to implement the instantiation of the triangle of reference [1] for the range of annotation tasks supported currently in the framework. The visualization in the Data Analytics component is developed using the Highcharts JS library. This component interacts with the CrowdTruth API, which is part of the Laravel framework. In the backend the requests are processed and translated into optimized aggregated queries for the MongoDB database. This protects the data stored into the database and optimizes the process, through efficiently querying the DB and partially executing in the backend the necessary computations for the raw response of the database. As a result, the number of computations in the frontend is reduced, the interface becomes more responsive, increasing the usability of the framework. On the other hand, the visual components are synchronized and communicate between themselves, e.g. general information and specific information views, as well as with their correspondent table views.

The visual components correspond to 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 (raw and annotated data, jobs, workers). 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. This enables quick addition and modification of charts, by changing the settings of the objects to be created. The configuration object of the charts empowers the easy extensibility of visualizations to new data types. Adding a new data type requires the specification of the properties to be visualized in the configuration object. Beside the barchart views, which are specific to each section (media, workers, jobs), all the other components of the views share the same implementation making the framework robust to changes and easily extendable.

The core of the CrowdTruth framework are the disagreement metrics [4, 3] evaluating the crowdsourced annotations in a variety of annotation settings, such as event extraction, video and image annotation, medical relation and factor extraction. Those 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. a 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. This is often quite simple, except for open-ended tasks. For each worker i submitting their solution to a micro-task on a MediaUnit u, the vector $W_{u,i}$ records their answers. The size of the vector depends on the number of possible answers per task. If the worker selects an answer, its corresponding component would be marked with '1', and '0' otherwise.

MediaUnit vector: For each task, we compute vector $V_u = \sum_i W_{u,i}$ summing across all workers. It accounts for all worker submissions on a media unit.

Below we describe the worker, unit and annotation metrics implemented in CrowdTruth framework. We show both their definition and examples of their visualisation in the CrowdTruth Analytics (see Figures 7, 8, 9).

5.5 Worker Metrics

The first metric below gives us a measure of how much a worker disagrees with the crowd on a unit basis, and the second gives us an indication as to whether there are consistently like-minded workers. The intuition is that there may be communities of thought that consistently disagree with others, but agree within themselves. Low quality workers generally have high scores in both.

Worker-unit disagreement is the average of all the cosine distances between each worker's Annotation vector and the full MediaUnit vector (minus that worker).

Worker-worker disagreement is $1 - avg(\kappa)$ for a particular worker. Since κ is a pairwise metric, we average, for each worker, the κ scores between that worker and all the others.

Average annotations per unit is measured for each worker as the number of annotations they choose per unit averaged over all the units (of a given AnnotationTask) 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 help indicate low quality workers. All three metrics are used to determine worker quality in the pie chart on the left In Fig. 7 and shown in In Fig. 8.

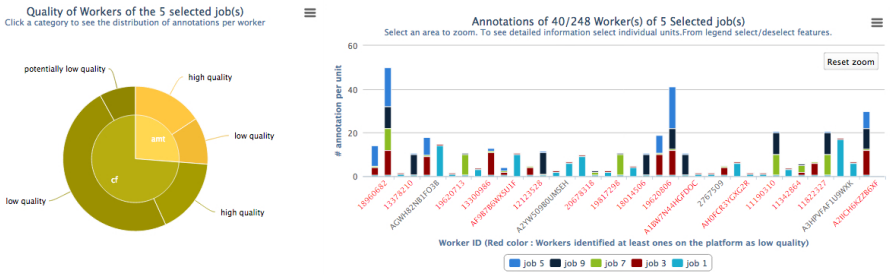

Fig. 7: Screenshot of CrowdTruth Analytics for Worker Quality and Annotations for Selected Jobs (comparison); click on a worker to se more details according to the CrowdTruth metrics & click on the pie chart to select specific jobs

5.6 Unit Metrics

Unit-annotation score, is the core crowd truth metric. It is measured for each annotation on each unit as the cosine of the unit vector for the annotation with the MediaUnit vector.

Unit clarity is defined for each unit as the max annotation score for that unit. If all the workers selected the same annotation for a unit, the max relation score will be 1, indicating a clear unit. Unit clarity is shown in Fig. 8, among

other worker and annotation metrics. This view is the most comprehensive tool to compare the performance of a sub-set of MediaUnits compared to the whole collection, or between each other.

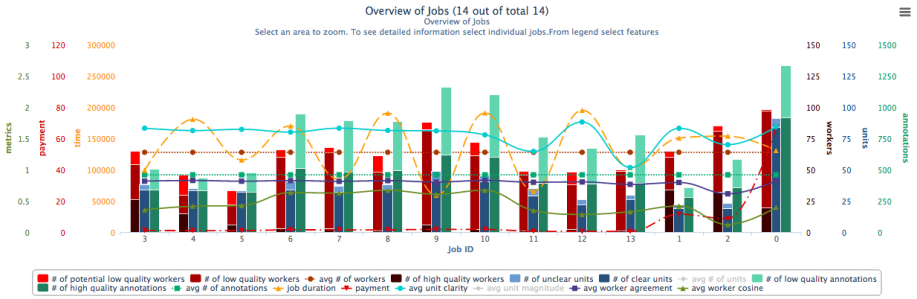

Fig. 8: Screenshot of CrowdTruth Analytics for Units in Selected Jobs and Tasks; click on an annotation bar for more details according to the CrowdTruth metrics & click on the pie chart to see annotations per micro-task

5.7 Annotation Metrics

Annotation similarity is defined as the *causal power* [7], 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 an 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 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.

Annotation ambiguity is defined for each annotation as the max annotation similarity for the annotation. If an annotation is clear, then it will have a low score. Annotations that are strongly associated with another may create problems for the annotation task, and for training machines 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, problems with the semantic specification, etc.

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. 9.

6 Related Work

The amount of knowledge that crowdsourcing platforms like CrowdFlower or Amazon Mechanical Turk hold fostered a great advancement in human computation [8]. 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]. 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] and

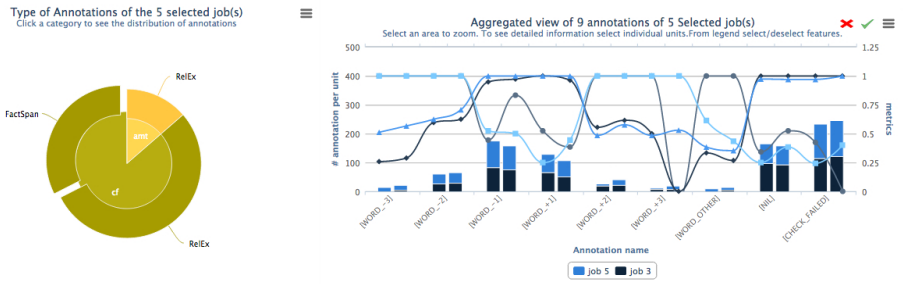

Fig. 9: Screenshot of CrowdTruth Analytics for Annotations on Selected Units in Selected Jobs; click on the pie chart to see the annotation distribution per micro-task

CrowdMap [10]. However, CrowdLang restricts the users to work with its own internal programming language and CrowdMap solves only ontology alignment. Thus, both frameworks can be hardly adapted to another domain.

A lot of research has been focused on identifying crowdsourced spam. Although a commonly used algorithm for removing spam workers is the majority decision [11], according to [12] it is not an optimal approach as it assumes all the workers to be equally good. Alternatively, expectation maximization [13] 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, [4] 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 [3] 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).

Although there is an extensive event extraction research using machines, the advantages of using crowdsourcing in this domain are not yet harnessed. Our new approach (fostering disagreement between annotators) [6] asks the crowd to judge the putative events and to provide event role-fillers at different granularities. The concept of harnessing disagreement in Natural Language Processing is not yet considered a mainstream process. In [14] disagreement is used as a trigger for consensus-based annotation in which all disagreeing annotators are forced to discuss and arrive at a consensus.

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 *groundtruth*, and is instead based on the observation that disagreement between annotators can signal low quality. The CrowdTruth methodology is based on the *triangle of reference* [1]. Thus, its implementation in the framework allows for easy adaptation to new micro-tasks. We have validated this, as the initial set of metrics was developed only for the medical text use cases of IBM Watson and

we easily applied to the new tasks for event and entity annotation in newspaper text, and for event annotation of videos and images.

We presented the details of the entire workflow together with the specifics of each framework component. We showed how such framework can be beneficial to the semantic web community with respect to the growing trend for crowdsourcing tasks, as well as with growing need for gold standard data. Detailed documentation, code and data export are provided online.

As future work, we would like to gather use cases from the semantic web community to extend the system with new data, micro-tasks and domains. Additional visualisations are also explored to increase the usability and effectiveness of the CrowdTruth metrics.

References

1. Aroyo, L., Welty, C.: Truth is a lie: 7 myths about human annotation. AI Magazine **in press** (2014)
2. Aroyo, L., Welty, C.: Crowd Truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. WebSci2013 (2013)
3. Aroyo, L., Welty, C.: Measuring crowd truth for medical relation extraction. In: AAAI2013 Fall Symp. on Semantics for Big Data. (2013)
4. 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)
5. 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)
6. Aroyo, L., Welty, C.: Harnessing disagreement for event semantics. In: Proc. of DeRiVE2012, ISWC2012. (2012) 31
7. Cheng, P.: From covariation to causation: A causal power theory. Psychological Review (104) (1997) 367âĂŞ405
8. 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
9. Minder, P., Bernstein, A.: Crowdlang-first steps towards programmable human computers for general computation. In: Human Computation. (2011)
10. Sarasua, C., Simperl, E., Noy, N.F.: Crowdmap: Crowdsourcing ontology alignment with microtasks. In: The Semantic Web–ISWC 2012. Springer (2012) 525–541
11. 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
12. 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)
13. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the em algorithm. Applied Statistics (1979) 20–28
14. Ang, J., Dhillon, R., Krupski, A., Shriberg, E., Stolcke, A.: Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In: INTERSPEECH, Citeseer (2002)