Development of a Crowdsourcing Multiagent System for Knowledge Extraction*

E. J. Azofeifa, G. M. Novikova

Peoples, Friendship University (RUDN University), Moscow, Russia

Abstract. Crowdsourcing is used for a wide variety of tasks on the Internet. From the point of view of knowledge extraction, it helps leverage knowledge in specific areas by gathering individual judgments of experts on specific subjects. In spite of crowdsourcing's proven effectiveness in tackling various sorts of problems, researchers do not coincide in a standard framework to represent and model this approach. In this work, a multiagent system (MAS) is presented as a method for modelling crowdsourcing processes intended to obtain expert knowledge. The system, exemplified by a corpus annotation process, includes a formulation of its goals in terms of uniqueness, value and temporality, and comprises a dynamic reward scheme that produces a real measure of inter-annotator agreement (IAA) while constraining the model to a time window and a reward limit.

Keywords: corpus annotation, crowdsourcing, expert knowledge, intelligent agents, dynamic reward, inter-annotator agreement.

DOI 10.14357/20718594200104

Introduction

Crowdsourcing is used for a wide range of tasks on the Internet. It consists of the transfer of a problem or task to a group of potential participants to solve it, and one of its most common uses is in the context of assessments or judgments. When the number of required evaluations of one of this kind of tasks, known as Human Intelligence Tasks (HIT), is too large for one expert, the combined assessment of a group is used to replace the expert's estimate. There are crowdsourcing platforms on which it is possible to hire specialists to perform these tasks. One of the most popular platforms is Amazon Mechanical Turk, a place where providers (requesters) offer a small monetary reward to people (workers / experts) in exchange of the completion of one or more small tasks (so-called micro-tasks). Crowdsourcing with micro-tasks, therefore, consists in providing a small monetary reward to a crowd per unit of work in short tasks (HIT).

Crowdsourcing is now used for a number of applications, such as classification and labeling of images, assessment of the quality of online content, identification of offensive or adult content, audio transcription, translation, product reviewing, transcription of scanned receipts, short paragraph spelling, mood analysis in tweets, peer assessment in online education, and so forth.

The quality of computer algorithms is often improved with the help of crowdsourcing with micro-tasks. Thus, the scalability of machines can be combined with large amounts of data, integrating as well the quality of human intelligence in its processing and understanding. Hybrid man-machine system models and paradigms have already been proposed at the tops of crowdsourcing platforms [1, 2], including the development of hybrid workflows. Among the examples of such hybrid human-

*This research was supported by the RUDN University Program 5-100.

✉ *Galina M. Novikova. E-mail: novikova_gm@mail.ru*

machine approaches are databases based on expert crowds, which use crowdsourcing to solve problems such as data integration, incomplete data, data aggregation, and graph search. Some information retrieval systems solve evaluation tasks using crowdsourcing. Expert crowds are used in Semantic Web systems for tasks such as object binding, schema mapping and ontology building.

In spite of crowdsourcing's proven effectiveness in tackling various sorts of problems, researchers do not coincide in a standard framework to represent and model this approach; instead, they often visualize such collective techniques either from a business point of view or from a technical one, with not so much in common between them. The former approach includes research focusing, for example, on the support of regular groups of workers [3] or the analysis of crowdsourcing supply elasticity based on historical data and models [4]. The latter approach involves the functioning of crowdsourcing systems under determined paradigms and constraints such as budgetary and latency limitations [5, 6].

Multiagent systems (MAS) have been proposed as a suitable formal approach to model collective behavior in crowdsourcing without leaving apart business factors [7], and with the advantage that existing multiagent techniques can be used to analyze individual behaviors in the crowd as well as trends in a macro level. In this work, the multiagent approach to crowdsourcing modelling is continued and reinforced with the addition of some considerations about knowledge extraction from a crowd of experts. In particular, the authors present a crowdsourcing model based on the example of corpus annotation, and outline its relation to knowledge characteristics such as temporality, value, relevance and uniqueness.

Details pertaining to the MAS are highlighted in the model, such as the different roles that intelligent agents can play in the crowdsourcing process as well as a suggestion about the agents' possible internal functioning. The proposed model details important factors from a technical perspective – such as the concept of inter-annotator agreement (IAA) [8] – but also their business counterparts in the form of cost, quality and time. The conjunction of those three factors in a MAS makes possible the visualization and development of crowdsourcing platforms capable of guaranteeing satisfactory results for all the parties within a predefined time window, which is especially valuable in fast or real-time crowd tasks such as weather forecasting, prediction markets, sports betting, some types of decision-making and time-constrained tasks in general.

1. Background

Tasks published by requesters in a crowdsourcing platform can be either simple (micro) or complex tasks. The former are characterized by requiring limited technical skills that are most likely possessed by a single worker, while complex tasks involves problem-solving techniques that usually require the coordination of groups of agents. This work focuses on the organization of individual intelligent agents (workers) for solving simple tasks under a marketplace model [9], which in contrast to contest and auction models usually requires a large number of solutions from a crowd of experts, each expecting a relatively small retribution.

The process of crowdsourcing with micro-tasks works as follows. First, the requesters design the HIT based on the required task and their data. Then they determine the specific requirements and the amount of money they are willing to pay the workers in exchange for the completion of each HIT. Afterwards, the providers publish each HIT in the corresponding part on the crowdsourcing platform. The workers/experts wishing to perform the published HIT(s) make a submission to the platform or directly to the requester. If accepted, the agents complete the task(s) and return their work to the provider. Finally, the requester receives the desired results and pays the experts accordingly (Fig.1).

The development of crowdsourcing applications is a process that includes several steps before and after the workers' resolution of tasks. In particular,

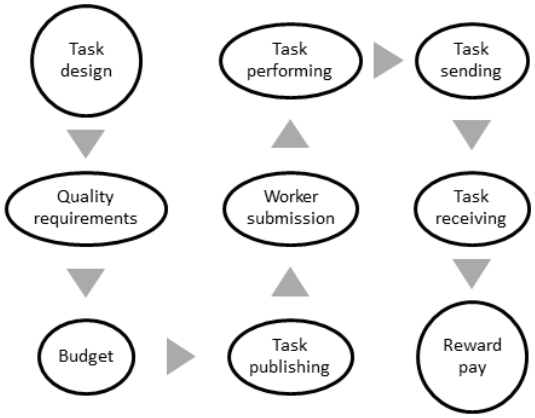

Fig.1. Overview of the process of crowdsourcing with micro-tasks

original tasks need to be decomposed into several levels of difficulty in a logical way, so that experts with different knowledge levels could perform them accordingly. In general, the process also includes steps such as workers' selection, rewarding and exclusion, the description of object-specific and worker-specific constraints, and the definition of a global cost and convergence criteria [10].

Non-expert evaluations of simple tasks, e.g. whether two products are the same or whether the image contains a face, are relatively easy and cheap to obtain with platforms such as Amazon MTurk. However, crowdsourced tasks involving more specialized knowledge can become a problem, since there is an increase in the difference between the cost of processing the preliminary data (performing the necessary actions to prepare the data for the task) and the cost of their evaluation (e.g. marking/annotation).

Detailed quality models have been proposed as frameworks to ensure that the pre-execution, execution and post-execution of the tasks comply with predefined minimum quality levels [11]. They often comprise methods to guarantee workers' expertise, often by means of a prescreening performed by the requester (Amazon MTurk, for instance, offers the ability to prescreen workers on a representative set of tasks) or by other classification means involving other experts, such as collective trust and reputation systems. Several approaches to the development of crowdsourcing platforms include methods for identifying the best workers in the crowd for specific tasks [12, 13].

Besides the expertise precondition, other key aspect in the development of crowdsourcing applications is the selection of a knowledge collection method that guarantees a certain reliability in knowledge extraction [14]. This objective can be attained by reducing noise, which in this context is defined as the lack of accuracy respect to the true (ideal) value for a task due to unintended actions or omissions.

Several scenarios in which noise can arise have been studied in the context of document labeling by [3]:

- Scenario one: labeling a very large collection of documents by means of a crowdsourcing platform leads to noisy answers: a big part of the answers is either almost correct (employees make honest mistakes), or arbitrary, (workers do not make any effort).
- Scenario two: expert estimates are subject to observation errors in the case where problems are associated with the analysis of indeterminate or nondeterministic data. This is because different subjects may give different values for the same problem due to a variety of reasons, such as different interpretation of scales, different wording of questions, and so forth.
- Scenario three: even if each task has an inborn or initial assessment, tasks related to the assignment of scores get different estimates from different reviewers/experts for reasons such as different content rating or different subjective interpretation of scoring scales.

On the other hand, not only unintended events are sources of inaccuracy. Considering a crowdsourcing process as an open MAS, agents (workers) are characterized as heterogeneous and self-interested, and their selfish behavior could lead to conflicting goals or malice [15]. Approaches to improve reliability under such MAS configurations have been studied, among them incentive mechanisms for workers to induce truthful behavior [16].

The effect of retribution on agents' work has been thoroughly studied in this context, and in general, researchers believe that a higher pay can move experts to deliver a higher quality work [14]. The basis of such approaches are models like Private Cost [17], which assumes that workers accept a task only if they can obtain a higher pay than the cost of performing the task, or Discrete Choice [5], which presupposes that agents will choose the tasks with the highest utility. In the context of crowdsourced design, for example, it has been found that raising the payments to the workers increases the probability of obtaining an outstanding solution from an expert, even though it does not improve the average creativity of the crowd [18].

Of particular importance are the attempts to optimize crowdsourcing reward mechanisms under certain constraints, and methods to determine appropriate paying amounts have been determined in several cases. For example, algorithms for defining optimal rewards have been developed in the context of decision making in the presence of noisy information [19] –where the amount paid affects the noise level–, and some noiseless frameworks allow the requester to set different requirements on the quality of experts' work in order to comply with a tight budget [20].

Depending on the quality requirements of the task, on the other hand, models have been created that dynamically increase or decrease the amounts paid to employees. They are tightly related to empirical observations of the way in which two types of

costs influence the individual quality of labels: those costs that are specific to the experts' operation (e.g. labeling), and those external or unrelated to it.

In these scenarios, the provider has the opportunity to collect only one opinion for each task in order to save money or other resource; however, trusting that opinion can lead to erroneous conclusions. A common practice to improve reliability is the use of redundancy: ensuring reliability by obtaining several markings or estimates (labels) for some or all data points. Recent crowdsourcing applications aim to improve accuracy by mixing skilled expert knowledge with varied levels of redundancy to tackle a possible lack of available knowledge. Authors have developed several applications based on this mix, among them an educational framework designed to create personalized curriculums subject to budgetary constraints [2].

The achievement of reliability in effective crowdsourcing systems with minimal redundancy costs has been studied; in particular, the development of algorithms that aim at obtaining the most accurate estimates from noisy evaluations given a defined amount of redundancy [3]. In this context, relabeling has been discussed as an instrument capable of directly improving the quality of the tagged data obtained by noisy but repeatable labeling, as well as the quality of the models derived from it.

Selective relabeling, the acquisition or distribution of labels/annotations on the basis of important factors to the provider, has been found to yield significant benefits [14]. This gives the possibility to develop methods based on a 'learning curve' approach that dynamically determine which action will give the highest marginal precision in the execution of a task. Such algorithms could evaluate in real time the acquisition of new examples and the selective relabeling of existing, noisy ones in order to calculate the expected benefit.

2. Crowdsourcing, MAS and Knowledge Extraction

2.1. A MAS as a Model for Crowdsourcing Processes

A multiagent framework will now be presented as the basis for a subsequent crowdsourcing modelling approach. Formally, the MAS can be described as a tuple *(T, A, M, O, L)*, where *T* denotes the set of tasks and *A = (R, W)* is a tuple conformed by the sets *R* of requesters (agents that offer payments in exchange of HITs) and *W* of workers (agents that perform the HITs to obtain a retribution). In this work, we refer to workers as 'agents' because, unlike requesters, they play an active role in our framework. However, the term can apply to both requesters and workers in more complex models where requesters could actively intervene in the functioning of the MAS. *M* denotes a collection of multiagent coordination mechanisms such as coalition formation, auction and negotiation (the minimal element of the set is the crowdsourcing platform); the set *O* corresponds to pre-execution, execution and post-execution operations such as task analysis, allocation, execution and feedback (results processing, rewards and learning), and *L* denotes the set of constraints over the operations.

Requesters are associated with tasks and tasks with subsets of operations by means of the respective relations $tr: R \rightarrow T$ and $ot: T \rightarrow P(O_z)$, where $P(O_z)$ denotes the powerset of $O_z \subset O$. Requesters make the tasks public so that agents in the set *W* perform them in exchange of a retribution. Using a determined utility model [17, 5], the agent performs a cost/benefit analysis, chooses one of the tasks, and sends a submission to the crowdsourcing platform or directly to the requester, which in turn accepts or rejects the submission.

If the request is accepted, the agent executes (evaluates) each operation of the task with an accuracy $ac: W \times O_z \rightarrow [0, 1]$, which defines a margin $\pm\omega_z ac(a,z)$ where the outcome of the operation (ω_z) is expected to be with a 100 % certainty. We assume that the execution of an operation gives as a result a number or can be represented as a number, and the outcome of a task is a linear combination of the results of its operations.

2.2. Agent's Acceptation or Rejection

However, the outcome of the task may change due to other factors; in particular, each combination of constraints in $L_z \subset L$, as well as the own effort of the agent, may influence the the result. Let $cs: W \times P(L_z) \rightarrow [0, 1]$ be a function that describes to which degree the precision of agent *a*'s evaluation is affected by each set of constraints. Then the following function yields the margin around the correct value for the task's outcome:

$$mr(a,t,l) = \frac{\varphi(\omega_1 ac(a,z_1),\omega_2 ac(a,z_2),\ldots,\omega_n ac(a,z_n))}{cs(a,l)}. \quad (1)$$

Where $\varphi(\cdot)$ is a linear combination (in this case, a weighted average), $z_1,z_2,\ldots,z_n \in ot(t)$ and $\omega_t = (\omega_1, \omega_2,\ldots, \omega_n)$. Equation 1 corresponds to the minimal margin that the agent can guarantee by applying a maximal effort $e_{at} = 1$, where $e_{at} \in [0, 1]$. We assume that an agent that makes zero effort performs evaluations with proficiency (probability of correctly calculating the outcome) $p_{\neq}(0) \leq 1/2$, and does no better than random guessing. If the requester allows for an error window of size ε_t, this means that the agent's expected outcome will randomly fall in an interval double the size of the error window:

$$mr^*(a,t,l) = \begin{cases} mr(a,t,l), & 0 < 2\varepsilon_t \leq mr(a,t,l) \\ (mr(a,t,l) - 2\varepsilon_t)e_{at} + 2\varepsilon_t, & 2\varepsilon_t > mr(a,t,l) \end{cases} \quad (2)$$

An agent who puts in full effort $e_{at} = 1$ attains its maximum proficiency. Putting in zero effort has a cost $c_{at}(0) = 0$, whereas putting in full effort has cost $c_{at}(1) \geq 0$. According to Equation 2, the probability $p_r(e_{at})$ of correctly calculating the outcome increases linearly with e_{at}, and it can be assumed that the cost $c_{at}(e_{at})$ does as well.

An agent is suitable to perform a defined set of tasks determined by the function $ta: W \rightarrow P(T)$, whereas the function $at: T \rightarrow P(W)$ yields the set of agents capable to solve a determined task. If we take $p_m(a,t)$ to be the probability that the margin $mr^*(a,t,l) \leq \varepsilon_t$, then the evaluation of the function for all $l \in P(L_z)$ can be considered as the condition of acceptance / rejection of an agent's submission to perform the task (Equation 3):

$$ar(a,t) = \begin{cases} accepted, & mr^*(a,t,l) \leq \varepsilon_t \; \forall \, l \in P(L_z) \\ rejected, & otherwise \end{cases} \quad (3)$$

If the quantity of constraints in L_z is considerably large, we can assume that there exists an ordering of the sets of constraints so that the difference $mr^*(a,t,l_i) - mr^*(a,t,l_{i+1})$ is negligible. This yields a smooth, differentiable curve in a 2D spectrum that we refer to as the ability of agent a for executing task t. In this case, Equation 3 can be substituted by the comparison of two curves -one associated to the agent and the other representing the minimal condition of acceptance of the task-, so that the agent will be accepted if the area under its curve comprises the totality of the area of the task.

2.3. Cost and Value for the Requester

An agent a's knowledge or ability can be unique in relation to other agents' knowledge. This is the case when there is a task $t \in T$ such that $|at(t)| = 1$, which has the potential to increase the cost for the requester. In this context, the authors consider the value of the obtained knowledge for the requester as a function $va: T \times [0, 1] \times G \rightarrow Q^+$, which yields a retribution for the requester depending on the evaluation accuracy of the tasks and the total time of the execution (G). This implies that perfect task accuracy is not sufficient to maximize the value for the requester: evaluations need to be performed within a defined time window that varies according to the goal of the requester.

For simplicity, it is assumed that finding $(t^*, \mu^*, g^*) = argmax(va(t, ac(t), g'))$ needs only a perfect execution of the tasks in a predefined time in the case of micro-tasks. However, the authors consider that the case of complex tasks calls also for an optimization of the pre-execution and post-execution, so that finding the maximum value for the requester becomes a search problem among all the possible combinations of the operations in the three processes. In this context, an increase in complexity can be related to semantically heterogeneous (interdisciplinary) tasks, implying that isolated knowledge from the experts –no matter their degree of specialization– could only take the requester's value to a local maximum. The authors consider that a global maximum value could be attained by delegating the rest of the knowledge extraction process –the pre and post-execution parts– to a crowd of interdisciplinary experts in order to prevent a loss of knowledge in the design, allocation and analysis of the tasks.

The presented MAS is assumed to be open [15], therefore functioning under conditions such as partial observability and decentralization. Concretely, agents, also characterized as heterogeneous and selfish, do not have knowledge about the totality of the environment due to cost constraints, and a central authority does not determine their behavior. For simplicity, we refer to the sets A and R as static, although the open MAS makes it possible to dynamically expand or reduce them in each time step.

3. A Crowdsourcing MAS Applied to Corpus Annotation

3.1. Presentation of the Model

A simple abstraction of the problem of developing mechanisms for knowledge extraction by means of crowdsourcing will now be presented. The model is based on the example of corpus annotation over text documents. A number of required annotations

(HITs), a total available time to perform all the annotations, and a total amount of money to pay the annotators define the annotation goal.

The system includes a mechanism that updates a dynamic HIT custom pricing scheme, in accordance to the annotation progress (the number of annotations obtained in relation to the available annotation time). As part of the pricing strategy, expert workers are directed to low-progress tasks by means of a bonus scheme. The model is defined in the following way.

Document: there are $|D| = n_d$ text documents, each of which is divided in a number $|S_d| = n_{sd}$ of sections.

Section: the length of each section s is l_s characters.

Tasks: there are m tasks, or objects, $j = 1,\ldots, m$, where each task has some underlying 'true quality' or type that corresponds to the ideal outcome ω_t around which the agents' evaluations must fall. In this model, the true type ω_j is related to only one curve of acceptance of the task, or q_j (known to the system), which we assume corresponds to a Gaussian curve whose parameters (height, width, center) range from 0 to 1 (exclusive). In order for an agent to perform a task, it must meet a minimum requirement, measured by the overlap $ar(i,j)$ between q_j and h_i, the agent's ability curve (also assumed to be Gaussian). For simplicity, all tasks should be performed on every section of a document.

Corpora: there are n_d different corpora. Each corpus is associated to a document and comprises m tasks, as well as the list U of annotations. There is a payment limit and a time limit, which constraint the payments that requesters are able to offer to experts.

Agents: the set W is conformed by n workers $i = 1,\ldots,n$ who noisily evaluate, or form judgments on, the qualities of objects. We say agent i performs task j if i evaluates object j. Agent i's judgment on task j is denoted by $ac(j) = y_{ij} \in [0, 1]$. We denote the set of tasks performed by an agent i by $T_i \subset T$, and let $A_j \subset W$ denote the set of agents who perform task j.

Proficiency: an agent's proficiency at a task j is the probability with which it correctly evaluates its true type or quality. Let h_{ij} denote agent i's ability, represented by a Gaussian curve with parameters u, v and $w \in [0, 1]$, and $pl: P(L_z) \rightarrow [0, 1]$ be the probability of encountering a determined set of constraints. Then the agent's proficiency can be calculated as follows:

$$pf(a,t) = \int pl(l) * ue^{-\left(\frac{l-v}{w\sqrt{2}}\right)^2} dl. \quad (4)$$

Equation 4 shows proficiency pf as the weighted integration of the Gaussian curve h_{ij} over the scale of possible sets of constraints. Agent i's minimum required ability depends on the task, and is enforced by the prescreening ta.

Strategies: depending on a probability of exploration (p_e), agents either choose one of the corpora they have worked with, or venture themselves to look for a new corpus. Once they define their corpus, agents strategically choose their tasks and, if their submission is accepted, they choose a certain effort level for each task in order to maximize their total utility (the difference between the reward received for their labeling or annotation and the cost incurred in choosing a task and making evaluations). Formally, an agent i's strategy is defined in a vector of tuples $S = [(f_{ij}, e_{ij})]$, specifying its effort level e_{ij} as well as the function f_{ij} it uses to choose a task in regard to other available options. The set S of effort levels and choosing functions is a full-information Nash equilibrium of a mechanism if no agent i can strictly improve its expected utility by choosing either a different function f_{ij}^* to choose its task or a different effort level e_{ij}^*.

Simulation: the model allows for agents (experts) to be generated either once at the beginning (static set A) or on each time step (dynamic set A), each of which automatically looks for a corpus to start annotating. Once it finds a corpus, the agent requests a task to perform. The corpus generates a list of rewards for each task, and presents it to the agent. Once the agent has chosen a task based on a cost/benefit analysis, the mechanism makes the prescreening ta of the agent according to the task (knowing the curve of acceptance q_j of a task). The prescreening ta depends on the calculation of the overlap between the Gaussian curve of the expert's ability and the curve of the task quality. If the agent meets the minimum ability and the maximum time (cost) to perform the task, it receives a confirmation and it starts the annotation. The agent gets the reward once it submits the annotation to the corpus. The corpus updates its task reward list according to the time elapsed and the annotation progress.

3.2. Reward Mechanism

The mechanism takes as input the set of all received annotations U_{ij} and computes a reward for each agent based on its annotations and the label-

ing of other agents. Rewards for each task are generated based on the annotation progress and the total projected annotation time (Equation 5).

$$G(x) = \frac{ch * m * a_p * r}{x}, x > r * m. \quad (5)$$

Where G is the time window, $ch = n_d * n_{sd} * l_s$ is the total number of characters in the document, m is the number of tasks (assuming every task must be performed on every section of the document), a_p is the difficulty factor associated to performing an annotation (where the base unit is the time needed to read a single character), r is the redundancy (number of replications of an annotation on a single section), and x is the total number of experts (high enough to avoid annotation duplicity of an expert on a single section). However, the total projected time depends on the available corpora on the system, which in turn modifies the total number of experts and turns the time window into $G^* = G(x_{nxt})$ (Equation 6).

$$x_{nxt} = x_{curr} - x_{out} + x_{in} = \\ = x_{curr}(1 - \overline{p_e}) + (x - x_{curr}) * \overline{p_e} * c'. \quad (6)$$

Where x_{nxt} is the number of experts in the corpus for the next time step t, x_{curr} is the number of experts working in the current corpus, $\overline{p_e}$ is the weighted average of the experts' p_e, and c' is the portion of time the current corpus is preferred over the rest of the corpora (due to its reward and bonus schemes). In other words, Equation 6 describes the inflow and outflow of experts in relation to the probability of exploring other corpora for better retribution conditions. Due to its dependence on experts' past choices, x_{nxt} cannot be established beforehand; instead, it must be calculated on each time step, so a safety factor of x can be used to mitigate possible variability.

A total available reward B is given to the system, as well as the percentage α of the reward that will be destined to low-progress annotation tasks (safety). Throughout the annotation process, the system updates the rewards either periodically or on every annotation done. At the moment of update, the system takes the proportion of time elapsed ($g1 - g0$) versus the total time available ($G(x_{nxt})$) as the mark for annotation progress. The mechanism then offers a bonus payment b_j for the annotation of blocks (tasks and sections) that progress slowly (below a mark β_j), in order to comply with the time window (Equation 7).

$$b_j = \begin{cases} b^*, & an(j) < \beta_j \\ b^* + \alpha * B * \frac{|an(j) - \beta_j|}{\sum_{t \in T'} |an(t) - \beta_t|}, & otherwise \end{cases} \quad (7)$$

Where $b^* = B^*(1-\alpha)^*c(a_p,j)$, the function $an:T \rightarrow [0,1]$ denotes the annotation progress, the function $c(a_p,j)$ corresponds to the reward weight of a task j depending on its difficulty, and T' is the set of tasks with underperforming annotation in terms of speed. Thus, the bonus is distributed among the low-progress blocks in a weighted manner, taking into account how far they are located in relation to the mark.

4. Discussion

The case of corpora annotation was chosen to illustrate the MAS due to its relative simplicity and its extensive use of micro-tasks. The agents in the model display a level of proficiency, which is determined by their innate ability (determined separately from other agents) and their choice of strategic efforts. However, workers in a potential simulation of the model can be set to perform up to their expertise, with no variability due to effort or motivation.

In order to guarantee data reliability (scenario one), noise in the model is reduced by fixed redundancy (a fixed number of annotations for every task): achieving this number of labels induces a minimal IAA, which in turn depends on the choice of a minimal acceptable degree of overlap between a worker's ability or expertise and a task's 'true quality'. Authors consider that the overall difficulty of an annotation is determined by the ratio of acceptance/rejection of experts applying to perform the task, and assume that the region not covered by the overlap of their abilities with the true quality of a task corresponds to observation errors (scenario two) and subjective interpretations (scenario three). The second scenario depends on the specific subject area of the task and is related to sensoring, while the third scenario could be tackled by pre-execution knowledge unification approaches such as the development of ontologies with their corresponding teaching strategies [21].

A dynamic reward system was built in order to constraint the model to a time window and a reward limit. This scheme is also applied to guarantee a fixed redundancy; however, the scheme can also be used as part of a selective relabeling strategy without a fixed redundancy, in order to maximize data reliability according to the IAA (or other) measure.

Annotations are assumed to comply to the quality requirements of the providers, because they are

enforced by a minimal expert acceptance threshold on every task. However, the total number of annotations for a task can be aggregated to obtain a real measure of IAA, on the base of which a more complex multiagent organization and reward system could be developed.

Both the pre-execution and post-execution processes can be transferred from the requester to a set of agents whose properties differ from the workers'. Concretely, a knowledge integration role can be assigned to experts with determined ability curves (high specialization and/or broad knowledge) in order to perform operations of task design, allocation (if the task requires it to be manual), and results aggregation or analysis. The latter could involve experts with broad knowledge generating new knowledge from the agents' work (e.g. annotations), possibly with the assistance of data mining or deep learning in hybrid approaches involving experts and artificial intelligence [22]. On the other hand, highly specialized agents can be selected to perform domain verification tasks, as it has been determined that such experts can outperform agents with broad knowledge in contexts constrained by time, budget or other resources [23].

Conclusion

Crowdsourcing is used for a very wide variety of tasks on the Internet. From the point of view of knowledge extraction, it helps leverage knowledge in specific areas by gathering individual judgments of experts on specific subjects. A concrete example is corpora annotation: the analysis of text documents by experts or qualified workers in the corresponding area, in order to extract implicit information from the texts.

In this work, a review of relevant literature has been presented, focusing on the use of crowdsourcing platforms to gather experts and obtain their knowledge on a wide array of subject areas. The concept of micro-task was considered as a base for the annotations in the concrete example, and strategies of noise reduction such as reward schemes and redundancy were reviewed. General considerations about multiagent systems have also been discussed, among them the formal expression of uniqueness, value and temporality of knowledge.

Finally, a MAS has been introduced as a method for modelling crowdsourcing processes intended to obtain expert knowledge. The system, which models a corpus annotation process, comprises a dynamic reward system that constraints the model to a time window and a reward limit. The MAS is designed to analyze an arbitrary number of experts, and model their interaction with different corpora. By aggregating the total number of experts' annotations, the system yields a real measure of IAA, whose reliability (noise reduction) is guaranteed by a fixed redundancy.

References

1. Kamar, E (2016). Directions in Hybrid Intelligence: Complementing AI Systems with Human Intelligence. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 4070–4073.
2. Azofeifa, E. J. & Novikova, G. M. (2018). VUZ: A Crowdsourced Framework for Scalable Interdisciplinary Curriculum Design. In Proceedings of IV International Conference on Information Technologies in Engineering Education (Inforino), Moscow, Russia, 1–6.
3. Karger, D. R., Oh, S. & Shah, D. (2013). Efficient crowdsourcing for multi-class labeling. In Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems. ACM.
4. Difallah, D. E., Catasta, M., Demartini, G., Ipeirotis, P. G., & Cudré-Mauroux, P. (2015). The Dynamics of Micro-Task Crowdsourcing: The case of Amazon MTurk. In Proceedings of the 24th International World Wide Web Conference (WWW 2015), pp. 238–247.
5. Gao, Y. & Parameswaran, A. (2014). Finish them!: Pricing algorithms for human computation. In Proceedings of the VLDB Endowment, *7*(14):1965–1976
6. Difallah, D. E., Catasta, M., Demartini, G. & Cudré-Mauroux, P. (2014). Scaling-up the Crowd: Micro-Task Pricing Schemes for Worker Retention and Latency Improvement. In Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP-14).
7. Jiuchuan, J., Bo, A., Yichuan, J., Donghui, L., Zhan, B., Jie, C. & Zhifeng, H. (2018). Understanding Crowdsourcing Systems from a Multiagent Perspective and Approach, *ACM* Transactions on Anonymous & Adaptive Systems, 13 (2), 8:1 - 8:32.
8. Nowak, S. & Rüger, S. (2010). How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In Proceedings of the international conference on Multimedia information retrieval (MIR '10), 557–566.
9. Ipeirotis, P. G. (2010). Analyzing the Amazon Mechanical Turk marketplace. *XRDS:* Crossroads, The ACM, Magazine for Students, 17 (2): 16–21.
10. Brambilla, M., Ceri, S., Mauri, A. & Volonterio, R. (2015). An Explorative Approach for Crowdsourcing Tasks Design. In Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion, 1125–1130.
11. Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., & Allahbakhsh, M (2018). Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. ACM Computing Surveys (CSUR), 51(1):7:1–7:40.

12. Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., & Vesci, G. (2013). Choosing the right crowd: Expert finding in social networks. In Proceedings of the 16th International Conference on Extending Database Technology, EDBT '13, pp. 637–648.
13. Difallah, D.E., Demartini, G. & Cudré-Mauroux, P. (2013). Pick-A-Crowd: Tell me what you like, and I'll tell you what to do. In Proceedings of the 22th International World Wide Web Conference (WWW 2013), ACM, New York, USA.
14. Sheng, V.S., Provost, F. & Ipeirotis, P.G. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD). ACM Press.
15. Huynh, T. D., Jennings, N.R. & Shadbolt N.R. (2006). An integrated trust and reputation model for open multi-agent systems. Journal of Autonomous Agents and MultiAgent Systems, 13(2):119–154.
16. Zhao, D., Li, X. & Ma, H. (2014). How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2014).
17. Singla, A. & Krause, A. (2013). Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In Proceedings of the 22nd International Conference on World Wide Web (WWW 2013), 1167–1178.
18. Wu, H., Corney, J. & Grant, M. (2014). Relationship between quality and payment in crowdsourced design. In Proceedings of the IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2014), 499–504.
19. Morrison, C. T. & Cohen, P. R. (2005). Noisy information value in utility-based decision making. In Proceedings of the First International Workshop on Utility-based Data Mining (UBDM'05), 34–38.
20. Li, Q., Ma, F., Gao, J., Su, L. & Quinn, C. J. (2016). Crowdsourcing high quality labels with a tight budget. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 237–246.
21. Gavrilova, T. (2010). Orchestrating Ontologies for Courseware Design. In Affective, Interactive and Cognitive Methods for E-Learning Design: Creating an Optimal Education Experience (Eds. by A. Tzanavari & N. Tsapatsoulis), IGI Global, USA, 155-172.
22. Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J. ... Horton, J. (2013). The future of crowd work. In Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13 (pp. 1301-1318). New York, NY, USA.
23. Novikova, G.M. & Azofeifa, E.J. (2016). Domain Theory Verification Using Multi-agent Systems. Procedia Computer Science, 120-125.