Community Profiling for Crowdsourcing Queries*

Khalid Belhajjame[1], Marco Brambilla[2], Daniela Grigori[1], Andrea Mauri[2]
[1] PSL, Paris-Dauphine University, LAMSADE, France
[2] Politecnico Di Milano, Italy

1. TELL ME WHAT YOU NEED, I'LL TELL YOU WHO TO ASK

Crowdsourcing has recently gained momentum as a means for outsourcing tasks to human workers with no specific qualifications or expertise. This is a viable option for tasks that are difficult to achieve correctly using machines or that are too expensive to be carried out by expert professionals. This leads to the possibility of addressing large campaigns of jobs without facing extremely high cost. While this technique has proven to be effective for simple labeling tasks, such as text recognition, the quality of results it delivers for relatively sophisticated tasks such as scientific paper translation, may appear as deteriorated. This is partly due to the fact that tasks get assigned to people who are not expert in the field. Therefore, there is a set of significant research challenges that still need to be addressed in crowdsourcing, especially on task-expertise matching considering issues like crowd platforms constraints, cross-platform crowd integration, and others.

Several researchers started elaborating on how to improve performance of the crowd. Some researches address the classical problem of cost vs. quality of results [2], by studying the trade-off between amount paid for every execution and overall quality, and by applying basic agreement or majority strategies over multiple executions of the same task [3]. Some other researchers study more complex patterns of tasks (e.g., create-fix-verify) that combined together allow for higher precision of results, see e.g., [5]. More recently, Venanzi *et al.* [6] and Li *et al.* [4] and proposed an algorithm for identifying communities. In doing so, they use statistical properties such as the reliability and confidence of workers and their demographics.

The objective of our work is instead to define in a rich way the concept of community of workers, defining a way to properly match crowdsourcing tasks to the communities based on expertise of workers and field/topic of tasks. We describe a set of conceptual models for queries, communities and workers, a high level architecture of the approach, and outline a set of possible strategies for addressing various aspects of expert-targeted crowdsourcing.

2. BUILDING COMMUNITIES OF EXPERTS FOR CROWDSOURCING

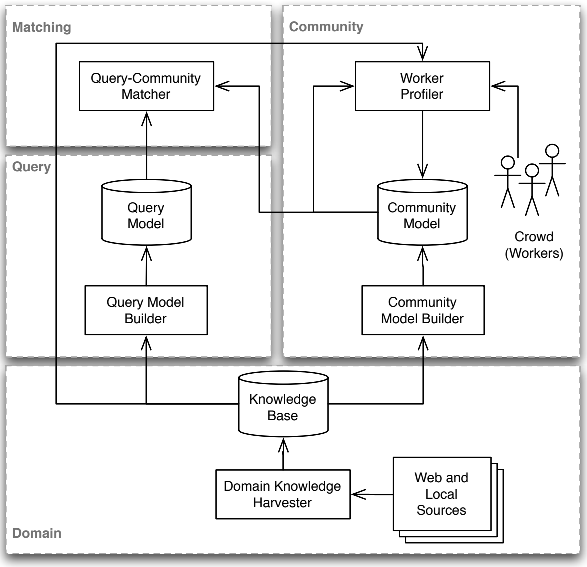

Figure 1: Overview of the proposed solution.

Figure 1 depicts the conceptual architecture we propose for addressing the problem of crowd targeting. The architecture covers four main aspects: Domain, Query (the question posed in the crowdsourcing task), Community and Matching. The harvesting component extracts information from existing web portals, social contents, etc., and stores it in a domain knowledge base to be used for describing communities and queries in the crowdsourcing campaign. In what follows we outline the models underlying the Community and Query aspects, and outline strategies for matching queries with communities.

2.1 Community Model

The community model is used to gather together workers characterized by an expertise potentially useful for answering given crowdsourcing queries. A community is defined intensionally in our system, meaning that it is not defined in terms of set of members, but instead in terms of its properties. Workers will be assigned to the community based on the matching of their profile with the community properties. In our model, we characterize a community using the following properties:

- Name, textual description, and tags (semantic annotations coming from the knowledge base) characterizing the community.
- Type of community: a community can be explicit, meaning

*The present work was carried out while Marco Brambilla was a visiting researcher at the Paris-Dauphine University.

that it is known among workers and it has a presence in the form, e.g., of a web portal or a social network group of interest; or implicit, i.e., determined purely by analyzing some domain knowledge harvested from the web.
• Duration: a community can be statically defined, reflecting an expertise that is potentially always needed. It can also be dynamic: some communities may emerge to respond to new needs, whereas others may disappear when they loose purpose.
• Grouping factor: the members of a community can be identified using properties such as interest, friendship, location, expertise, affiliation, etc.
• Communication channel: communities need a medium to be engaged. The communication channel refers to the means by which the members of the community can be solicited. For example, Facebook, Linkedin, Twitter, blogs or web sites (reviews, expert sites), Amazon Mechanical Turk, etc.

Building Communities.

To define communities, the community builder can harvest multiple sources of information, such as the knowledge base itself, previous crowdsourced queries and worker profiles. The builder is also in charge of defining relationships between communities. We distinguish between two kinds of relations: subsumption and similarity. Subsumption is used to specify that a given community comprises another community. Similarity is used to specify that two communities refer to similar topics or expertise. For example, communities of experts on classical music and on opera can be considered similar (to some extent, in the range 0 to 1).

Profiling Workers.

The workers that join (or are associated with) a given community are profiled using one of the following methods: i) Explicitly, i.e., by asking the workers to identify his/her expertise from a list of kinds of expertise or concepts; ii) Implicitly using their profile information on the crowd platform, when available; iii) Implicitly using work quality, i.e., by asking the worker a list of questions, and identifying his/her expertise by analyzing the quality of results (assuming to have the ground truth of the questions), possibly including tasks performed in the past, when available. Note that a worker may be associated with more than one community, meaning that s/he posses multiple kinds of expertise.

2.2 Query Model

The query model describes the queries, i.e, the questions asked to the crowd within a tasks, for which executors are required to provide a response. We distinguish between query template and concrete query. A query template defines the structure of the question to be asked to the worker, without referring to specific entities, whereas a concrete query refers to particular entities. An example query template is “Was movie X directed by director Y?” (note that it does not specify the movie or the director in question). A concrete query instance of such a template would be, e.g., “Was Titanic directed by James Cameron?”. A query template is characterized by a textual description of the task it involves, the kind of operation that is requested to the worker, and the definition of expertise that may be needed for the worker to perform it (for instance, expertise on movies). A catalog of kinds of operations is defined [1] and describes the job the worker will be asked to perform (e.g., classifying, ranking, tagging, or liking the object of interest). Concrete queries can also include more precise description of needed expertise (for instance expertise on romance or thriller movies). Therefore, the whole query model is annotated with concepts from the domain knowledge base. Finally, a query may be also temporal-dependent, meaning that the correct response may change over time.

To assist the user in specifying a query model, the query builder model provides information in the knowledge base that may be of assistance, such as tags or keywords that are used and understood by the crowd and that can be matched with the worker profiles.

2.3 Matching

Given a query (be it template or concrete) and a collection of communities, the role of matching is to associate the query to the communities of workers that are best suited for the task implied by the query. We envisage the following two matching strategies.
• **Naive keyword-based matching.** Communities and queries are treated as bag of words. The bag of words are extracted from the properties characterizing the communities (name, description, etc.) and the query. Matching is calculated as the overlapping of the two bags of words.
• **Semantic matching.** Communities and queries are mapped to concepts (tags, taxonomies) described in the domain knowledge base, which capture the expertise they provide and require respectively. The matching is then performed based on such semantic annotations, also considering semantic associations between them.

Both strategies have their pros and cons. Keyword-based matching is simpler, however, it does requires indexing of content and will likely perform poorly since it does not consider only the expertise, but all other terms appearing in the descriptions. The semantic strategy is more promising, since it focuses on concepts describing the expertise, but it requires to assert or infer semantic annotations characterizing both communities and queries, and their similarity.

3. OUTLOOK AND RESEARCH AGENDA

We believe the field of expert-based crowdsourcing is still largely unexplored and open to innovation. We presented a high-level description of our vision that aims at improving the quality of crowdsourcing results through expertise matching, without requiring increase of cost, thanks to automatic community building and matching techniques. Our analysis gave rise to several research issues, including:
• how to harvest a knowledge base that allows profiling communities and user queries in an optimal way.
• how to cope with the dynamics of both communities and work, due to changing needs, communities and expertise of workers over time.
• how to deal with query requiring multiple kinds of expertise, or expertise that is not explicitly defined within the community model.

We intend to address the above issues and others in our work, and invite others from the information and knowledge management community to join us in these challenges.

4. ACKNOWLEDGMENT

We would like to thank the Keystone COST Action IC1302, which supported the work reported in this paper.

5. REFERENCES

[1] A. Bozzon, M. Brambilla, S. Ceri, and A. Mauri. Reactive crowdsourcing. In *WWW*, pages 153–164. ACM, 2013.

[2] G. Demartini, D. Eddine Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, pages 469–478, 2012.

[3] A. Gruenheid and D. Kossmann. Cost and quality trade-offs in crowdsourcing. In *DBCrowd*, pages 43–46, 2013.

[4] H. Li, B. Zhao, and A. Fuxman. The wisdom of minority: discovering and targeting the right group of workers for crowdsourcing. In *WWW*, pages 165–176. ACM, 2014.

[5] P. Minder and A. Bernstein. Crowdlang: A programming language for the systematic exploration of human computation systems. In *SocInfo*, pages 124–137. Springer, 2012.

[6] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *WWW*, pages 155–164. ACM, 2014.