When Crowds Give You Lemons: Filtering Innovative Ideas using a Diverse-Bag-of-Lemons Strategy

IOANNA LYKOURENTZOU, Utrecht University, Netherlands
FAEZ AHMED, University of Maryland, United States
COSTAS PAPASTATHIS, University of Peloponnese, Greece
IRWYN SADIEN, Research Institute of Big Data Analytics, Department of Computer Science and Software Engineering, Xi'an Joaoton-Liverpool University, China
KONSTANTINOS PAPANGELIS, Research Institute of Big Data Analytics, Department of Computer Science and Software Engineering, Xi'an Joaoton-Liverpool University, China

Following successful crowd ideation contests, organizations in search of the "next big thing" are left with hundreds of ideas. Expert-based idea filtering is lengthy and costly; therefore, crowd-based strategies are often employed. Unfortunately, these strategies typically (1) do not separate the mediocre from the excellent, and (2) direct all the attention to certain idea concepts, while others starve. We introduce DBLemons – a crowd-based idea filtering strategy that addresses these issues by (1) asking voters to identify the worst rather than the best ideas using a "bag of lemons" voting approach, and (2) by exposing voters to a wider idea spectrum, thanks to a dynamic diversity-based ranking system balancing idea quality and coverage. We compare DBLemons against two state-of-the-art idea filtering strategies in a real-world setting. Results show that DBLemons is more accurate, less time-consuming, and reduces the idea space in half while still retaining 94% of the top ideas.

CCS Concepts: • **Information systems** → *Crowdsourcing*; *Information retrieval diversity*; • **Human-centered computing** → *Collaborative and social computing systems and tools*;

Additional Key Words and Phrases: open innovation, diversity, filtering

ACM Reference Format:
Ioanna Lykourentzou, Faez Ahmed, Costas Papastathis, Irwyn Sadien, and Konstantinos Papangelis. 2018. When Crowds Give You Lemons: Filtering Innovative Ideas using a Diverse-Bag-of-Lemons Strategy. 1, 1 (August 2018), 23 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Open innovation is the process where an organization opens up to external parties (customers, stakeholders, volunteers) to gather out-of-the-box ideas on how to solve challenging problems. The rationale is that while many problems can be solved within the traditional boundaries of the firm, sometimes the knowledge, intuition and radical creativity required for solving new problems is

Authors' addresses: Ioanna Lykourentzou, Utrecht University, Netherlands, i.lykourentzou@uu.nl; Faez Ahmed, University of Maryland, United States, faez00@umd.edu; Costas Papastathis, University of Peloponnese, Greece, cst12079@uop.gr; Irwyn Sadien, Research Institute of Big Data Analytics, Department of Computer Science and Software Engineering, Xi'an Joaoton-Liverpool University, China, Irwyn.Sadien@xjtlu.edu.cn; Konstantinos Papangelis, Research Institute of Big Data Analytics, Department of Computer Science and Software Engineering, Xi'an Joaoton-Liverpool University, China, KonstantinosPapangelis.

not available and must be sought outside. In the last decade, open innovation has led to a major shift in how we think about R&D: from siloed, in-house discovery to the engagement of external crowds, with leading firms (like Intel, Cisco [21] and Lego [34]), semi-autonomous organizational collectives [20], and innovation brokers (OpenIDEO, Innocentive[1]) all relying on it to select the projects to be funded.

However, one of the main – and persistent – problems that organizations face after an idea contest is that of filtering. Contributors have just sent in a flood of candidate solutions of variable quality, and these solutions must now be reviewed, and the most promising among them must be selected. At this stage, organizations usually rely on in-house experts who will evaluate and filter the ideas. This can be a cumbersome, costly and lengthy process, which creates significant production bottlenecks and increases the transaction costs for searching and evaluating externally-sourced knowledge [25]. An indicative example is Google's 10 to the 100^{th} project, which received over $150,000$ suggestions on how to channel Google's charitable contributions [47]. To deal with the unexpected deluge of submitted ideas, Google had to allocate 3000 employees for the filtering of the ideas; a process that put them nine months behind schedule. Furthermore, research has shown that the in-house experts may miss good solutions due to the significant cognitive load involved in reviewing multiple diverse ideas in a short time frame [46, 48]. Together, these problems cause serious concerns about the practical usability of open innovation and *often make organizations, as well as investors reluctant about using open innovation altogether* [24, 30].

In view of the above, researchers have recently started to explore using the crowd to filter the candidate ideas. The most typical strategy used by popular open innovation platforms like OpenIDEO is majority voting, where crowd raters go through the candidate ideas and upvote their preferred ones, and ranking is dynamically[2] calculated in a descending vote order. The problem with this strategy is that it is prone to quickly locking into a fairly static and arbitrary ranking of the ideas because of positive feedback loops, as people tend to fixate on the few ideas that have already received good ratings or are readily visible [43]. For example, in the OpenIDEO challenge with which we will work on in this paper, half of the crowd's evaluations went to only 10% of the ideas, while one fifth of the ideas (21%) received no evaluation[3]. A second problem with majority voting is that the crowds are less effective in distinguishing mediocre from excellent ideas [29].

To address these issues, in this paper we propose DBLemons: a crowd-based idea filtering strategy which helps increase filtering efficiency by balancing idea quality and idea concept space coverage. We compare DBLemons against two ranking strategies: i) majority voting, which replicates the standard voting mechanism used in today's online innovation communities and ii) Bag of Lemons [29], a state-of-the-art approach that uses negative instead of positive voting, which we extend for use in the dynamic ranking setting of real-world innovation platforms.

Our work makes three main contributions. We show that:

- Balancing idea quality and idea representativeness improves the filtering of high-quality ideas, as it helps people make better comparative decisions.
- Majority voting is more time consuming and less accurate than the other two alternatives.
- Bag of Lemons with dynamic ranking has similar filtering efficiency as Majority voting, but finds good ideas faster. This result confirms Klein and Garcia(2015) on the dynamic setting.

The rest of this paper is structured as follows. In section 2 we analyze related work and the different approaches of idea evaluation, ranking and filtering. In section 3 we present our methodology,

[1]https://openideo.com/, https://www.innocentive.com/
[2]Dynamic ranking means that the number of votes per idea is updated every time a user casts a vote, and this information is used to re-calculate the ranking that a new user sees when he/she first enters the platform. We will use this definition throughout the paper.
[3]OpenIDEO challenge on women's safety: https://challenges.openideo.com/challenge/womens-safety/refinement, Evaluation stage

comprising dataset creation and the three ranking strategies, including DBLemons. We continue with section 4 and the presentation of our experimental results, which illustrate the efficacy of our method when applied to a real-world idea filtering problem. Next, in section 5, we discuss these results in light of their impact on open innovation, present limitations and future work. Finally, in section 6 we conclude with the main findings and take-away messages of the paper.

2 RELATED WORK

2.1 Idea Evaluation: Who will be the reviewer?

Machines or Humans. Ideas can be judged using machines, humans or a combination of the two. Machine rating is generally applied to ideas that are in structured format. For example, ETS grading [16] uses natural language processing to grade papers. However, it is still difficult for machines to evaluate ideas on aspects like creativity, as this requires combining high-level knowledge from heterogeneous sources. In contrast, human intelligence excels at acquiring, understanding and making mental connections among diverse knowledge sets, and in making abstractions. Although very recent literature has moved towards the direction of teaching machines how to perform these tasks [23], *humans are still the best option for intellectual, subjective tasks like innovation assessment.*

Experts or crowds. Human evaluators can be experts or non-experts. Experts have a substantial knowledge of the field and of the market, and can thus provide more informed and trustworthy evaluations [12]. Many crowdsourcing platforms such as Topcoder, Taskcn, and Wooshii use expert panels to select contest winners [11]. However, experts are also scarce and expensive, since gaining expertise on a particular innovation subfield takes a substantial amount of training. In practice, convening an expert group to evaluate the large number of ideas of an open innovation contest has proven to be prohibitively slow, costly, and to cause significant decision and production bottlenecks [28]. Crowds have been proposed as an alternative to evaluating ideas that require human input. Apart from the evident advantage of being faster and more cost-effective [32], adequately large numbers of people have proven to make accurate estimations of reality due to their large diversity of viewpoints, knowledge and skills ("wisdom of the crowds" notion [45]). Expertise can also be found within the crowd, and researchers have searched for ways of leveraging it through expertise-weighted consensus mechanisms [9], and priming techniques for novices to serve as expert proxies [37]. Under the right conditions, crowd-based idea evaluation has been shown to be equivalent to that of experts [26]. Evidence finally exists that crowds can provide high quality opinions on difficult judgment and choice tasks, frequently outperforming individual experts [17]. *Overall, recent literature shows that crowds have potential in evaluating promising ideas. Combined with the prospect for reduced cost and faster turnaround times, they constitute an alternative that needs to be explored for high-quality idea filtering.*

2.2 Idea Evaluation: How will the ideas be selected?

Author-based or content-based. Idea evaluation mechanisms can be broadly classified into author-based or content-based [27]. In author-based evaluation, the reputation of the author is the main determinant of selecting good ideas. This reputation may be known a-priori or it may be gradually assessed over longitudinal tasks [8]. Ideas from reputable authors or authors scoring high on standardized questions are selected. Content-based filtering places focus on the idea itself and its potential to effectively solve the given problem. This mechanism can be further divided into two evaluation mechanisms: i) judgment-based or ii) choice-based.

Judgment-based or choice-based. Judgment-based evaluation involves individually assessing each idea on a pre-agreed rating scale (e.g. Likert) [15, 40]. It offers the advantage of a meaningful interpretation of the result, since all ideas are assessed against an absolute standard (the scale's endpoints). It has also been connected to a higher perceived ease of use and higher decision quality [7]. However, because each idea is assessed individually, judgment-based evaluation takes time and can thus be expensive [28]. *In this paper we will use the judgment-based approach to create the benchmark dataset of ideas, which we will use to compare the three crowd-based filtering strategies that the paper investigates.*

Choice-based filtering involves comparing a set of ideas and then, in accordance to certain evaluation guidelines, selecting some of these ideas based on one's own preference [35, 39], or based on the expected preferences of other evaluators [7]. Because it involves comparisons and not individual assessment, choice-based filtering is faster and thus more cost-effective for use by large numbers of evaluators [50]. However, since it does not evaluate ideas against an absolute criterion, its performance is affected by the ranking strategy, i.e. the order in which the ideas are presented to the crowd. *In this paper we propose a new choice-based filtering strategy and will compare it with two real-world and state-of-the art alternatives.*

2.3 Idea Ranking and Filtering

Majority voting. In majority voting, each voter gets to upvote ideas they like (similar to Facebook "Likes"). Ideas are ranked in descending order based on the number of upvotes they receive. This method enables a crowd to provide similar accuracy to a Likert scale-based evaluation but at a fraction of the required time [28]. However, it also presents two problems when applied to large idea collections. First, the "snowball" effect, where voters are fixated on a few ideas or idea concepts (ideas with similar thematic) because these ideas or concepts received the first initial upvotes and thus are more likely to be seen (and voted on) by subsequent users, while other potentially better ideas do not receive an equal share of attention [43]. Second, when positive voting is used, as in majority voting, users are less likely to distinguish mediocre from excellent ideas [29]. Despite the above, majority voting is a straightforward method for voters to understand, and thus it is widely used by many online open innovation communities like OpenIDEO and GrabCAD [43].

In this paper we will replicate majority voting and use it as a baseline for comparison with the other two ranking strategies with which we will experiment.

Bag of Lemons. To address the second problem of majority voting, i.e. the difficulty of crowd voters to filter mediocre from excellent ideas, Klein and Garcia (2015) introduced the notion of "Bag of Lemons" (BoL). The key insight is that crowds are better at eliminating bad ideas than they are at identifying good ones. In their experiment, a group of lab members were informed that a given set of ideas had been reviewed by an expert committee, and that their job was to predict which ideas had been selected as winners. They used the BoL multi-voting technique with static ranking (ideas are not re-ordered as participants vote on the platform) and compared these results with the Bag of Stars and Likert scale approaches. They found that using the Bag of Lemons approach provided a better recall/compression trade-off with significant time improvements. Other literature on BoL [50] has looked into user activity when using BoL and compared it to both the Likert scale and up/down-voting.

In this paper we extend the BoL approach on a *dynamic voting setting*, where multiple voters arrive at different times, with unknown arrival rates, and where each voter views the ideas ranked according to the votes of the previous users. In this setting, which is closer to the actual conditions of real-world open innovation communities, we explore whether Bag of Lemons will still manage to increase filtering efficiency and reduce task time for the crowd workers.

2.4 Cross-Domain inspiration: Idea Diversification

To address the first problem of majority voting, i.e. that voters tend to be fixated on a few idea concepts while others receive disproportionately less attention, we draw inspiration from the notion of *diversity.* Diversity is most often encountered in the fields of information retrieval and recommender systems, where researchers seek to recommend interesting sets of items to people (*e.g.*, movies on Netflix), and where predicting exactly what the user wants is difficult. One strategy around this is to recommend a diverse set of items, hoping that by covering a diverse space of options, the chances of matching one of the recommended items to user preferences will increase. The intuition for this approach stems from the *portfolio effect* [5] where placing similar items together has decreasing additional value for users. This *diminishing marginal utility* property is also well-studied within consumer choice theory and related fields [13]. Various approaches have been proposed for representing and optimizing this diminishing marginal utility to achieve efficient diversification [49]. Such approaches relate to a broad set of applications like music discovery [52], keyword-based summarization [18], ecology[38], and document summarization [53]. *In this paper, we examine if diversifying the idea ranking based on thematic concept clusters (sets of ideas with similar thematic) and combining this with the BoL strategy (which has proven to be better than majority voting at least on static settings) will ensure a better coverage of the idea space and thus help increase filtering efficiency.*

In summary, building on related literature, in this paper we aim at answering the following two research questions:

- Does Bag of Lemons (BoL) outperform majority voting in filtering efficiency within a dynamic vote ranking setting?
- Does diversity-assisted ranking increase BoL's efficiency in filtering high-quality ideas?

3 METHODOLOGY

We first create a dataset of a real-world open innovation problems. In this dataset, we compare the three idea ranking and filtering strategies: i) Majority voting, ii) Bag of Lemons (BoL) and iii) Bag of Lemons with idea diversification (DBLemons).

3.1 Dataset Creation

3.1.1 Real-world idea selection and summarization. To test open innovation idea filtering we need a dataset containing real-world ideas. A straightforward solution would be to use ideas from an existing open innovation problem and test the new algorithms on the top ideas selected for this problem by experts or by the crowd. This approach has two problems. One, by asking crowd workers to vote on ideas, the text and ratings of which are publicly available, one risks that crowd workers will simply look these ratings up and provide biased evaluations. Two, it is difficult to ascertain if the top ideas selected by the crowd through an existing open innovation platform were selected based solely on merit or if this selection was affected by other factors, like word count and number of comments as previous research indicates [2]. Hence, we decided to generate a new dataset. We proceeded as follows.

First, to make our dataset as close to reality as possible, we gathered a set of ideas posted by community members of a successful online innovation platform, called OpenIDEO. OpenIDEO promotes social impact by designing products, services and experiences that build on the ideas of its distributed community [19]. It hosts idea "challenges" around social issues. Each challenge has four stages: i) Research, ii) Ideas (hundreds of thousands of idea submissions), iii) Evaluation (filtered subset of ideas, 10% of the previous stage), and iv) Winners. To browse through or upvote ideas, OpenIDEO users can order the ideas by date, total number of comments, or total applauds, which are gradually accrued over time. Past work has investigated finding a smaller subset of diverse ideas on OpenIDEO and training classifiers to rank ideas by quality [4] [2]. It has also been shown that the top-rated ideas in OpenIDEO are in the bottom 5 percentile of diversity, meaning that the top ideas shown are very similar to one another [3].

We created our dataset by summarizing ideas from the Evaluation stage of an OpenIDEO challenge on women's safety[4]. The decision to work with the ideas of the Evaluation, rather than the Ideation stage was taken for two reasons: i) all Evaluation-stage ideas have a minimum quality and structure, compared to ideation stage ideas which may be of poor quality or stubs, ii) summarizing 600 ideas is time consuming and not central to the research question that is the focus of this work. The chosen challenge called for ideas to solve the following problem:

"How might we make low-income urban areas safer and more empowering for women and girls?"

We summarized each idea in approximately 150 words, taking care to remove identifying information that could lead back to the original OpenIDEO idea description. Each summary was reviewed sequentially by 3 reviewers of the author team to homogenize the writing style and avoid bias due to different writing skill levels. In the end, we acquired a dataset of 52 idea summaries.

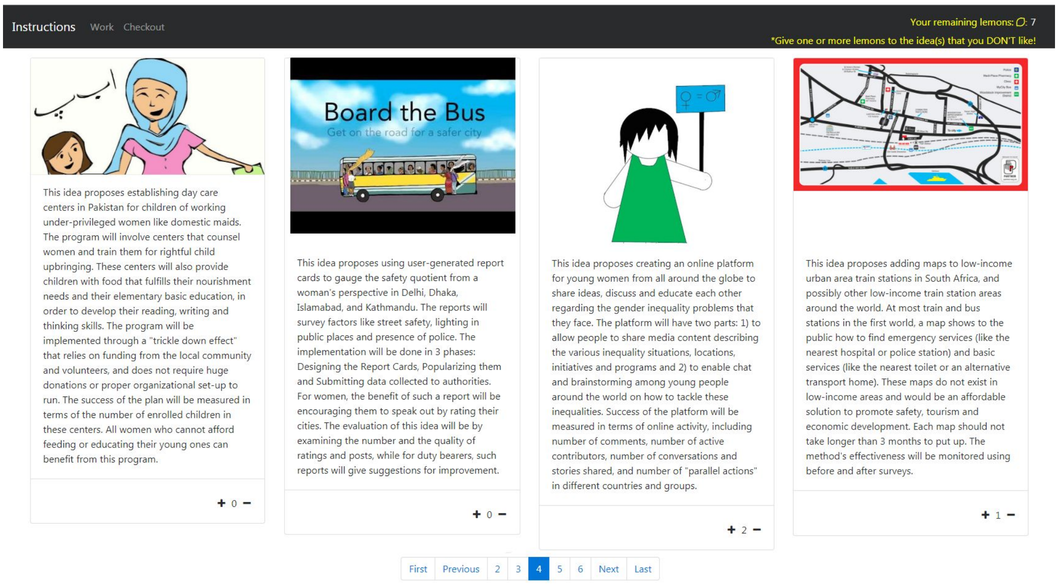
3.1.2 Dataset Evaluation. The next step is to evaluate our dataset, and identify the subset of top ideas (hereby called "golden set") that will be used to compare the ranking strategies. Using the judgment-based idea evaluation approach, we use crowd ratings to evaluate each idea. We hired a total of 520 Figure Eight[5] (previously named CrowdFlower) workers and asked each of them to evaluate three ideas on a 5-point Likert scale on the following quality axes: i) Investment potential, ii) Novelty, iii) Impact potential, iv) Feasibility, v) Scalability, vi) Understandability and vii) Overall feeling. The axes were selected in accordance with the common axes used by OpenIDEO to evaluate its ideas in different challenges. In the end, each idea was evaluated by 30 crowd workers. Using the average ratings across axes and workers, we obtained the final quality score for each idea. Using these scores, we selected the Top 30% (16 ideas; a similar selection ratio to that of OpenIDEO, which for this challenge selected 15 finalists out of the 52 ideas). These ideas constitute our golden set, over which we will compare the three ranking strategies of our experiments.

3.2 Ranking strategies

3.2.1 Majority Voting. Majority voting replicates the standard voting mechanism used in online design communities. Each rater gets up to 52 votes (the size of our idea dataset). They are free to use them to upvote any number of ideas but they cannot allocate more than one vote per idea. A rater sees the ideas by visiting our idea platform (the functionality of which we present in detail later on, in the Experimental Setup sub-section). When the rater visits our platform, ideas are sorted in descending order by the total number of votes they have already received (i.e. ideas with the most votes go at the top). Dynamic ranking is used, i.e. the number of votes per idea is updated every time a user casts a vote, and this information is used to re-calculate the ranking that a new user sees when he/she first enters the platform.

3.2.2 Bag-of-Lemons (BoL). In this strategy we adopt the Bag of Lemons approach proposed in literature and combine it with dynamic ranking. Each participant is given a budget of 10 "lemons", and they are asked to distribute them to the ideas they feel are the least likely to be selected as winners by an expert committee (the actual winning ideas are kept secret from the workers until the end of the task). The focus here is on eliminating bad ideas, rather than identifying good ones.

[4]https://challenges.openideo.com/challenge/womens-safety/refinement
[5]https://www.figure-eight.com/



This idea proposes establishing day care centers in Pakistan for children of working under-privileged women like domestic maids. The program will involve centers that counsel women and train them for rightful child upbringing. These centers will also provide children with food that fulfills their nourishment needs and their elementary basic education, in order to develop their reading, writing and thinking skills. The program will be implemented through a "trickle down effect" that relies on funding from the local community and volunteers, and does not require huge donations or proper organizational set-up to run. The success of the plan will be measured in terms of the number of enrolled children in these centers. All women who cannot afford feeding or educating their young ones can benefit from this program.

This idea proposes using user-generated report cards to gauge the safety quotient from a woman's perspective in Delhi, Dhaka, Islamabad, and Kathmandu. The reports will survey factors like street safety, lighting in public places and presence of police. The implementation will be done in 3 phases: Designing the Report Cards, Popularizing them and Submitting data collected to authorities. For women, the benefit of such a report will be encouraging them to speak out by rating their cities. The evaluation of this idea will be by examining the number and the quality of ratings and posts, while for duty bearers, such reports will give suggestions for improvement.

This idea proposes creating an online platform for young women from all around the globe to share ideas, discuss and educate each other regarding the gender inequality problems that they face. The platform will have two parts: 1) to allow people to share media content describing the various inequality situations, locations, initiatives and programs and 2) to enable chat and brainstorming among young people around the world on how to tackle these inequalities. Success of the platform will be measured in terms of online activity, including number of comments, number of active contributors, number of conversations and stories shared, and number of "parallel actions" in different countries and groups.

This idea proposes adding maps to low-income urban area train stations in South Africa, and possibly other low-income train station areas around the world. At most train and bus stations in the first world, a map shows to the public how to find emergency services (like the nearest hospital or police station) and basic services (like the nearest toilet or an alternative transport home). These maps do not exist in low-income areas and would be an affordable solution to promote safety, tourism and economic development. Each map should not take longer than 3 months to put up. The method's effectiveness will be monitored using before and after surveys.

Fig. 1. Testing platform screenshot, BoL/DBLemons strategies

Ideas are ranked in ascending order of the total number of lemons they have received (ideas with the least number of lemons, i.e. of higher quality, are at the top). Dynamic ranking is used, in contrast to the Bag of Lemons approach in Klein and Garcia (2015). The choice of 10 lemons was selected similar to that work for a dataset of similar size (50 ideas in Klein and Garcia (2015 and 52 in our dataset).

3.2.3 Diverse Bag-of-Lemons (DBLemons). In this strategy we combine the notion of diversity with the Bag of Lemons approach. Similarly to the BoL strategy, each participant is given 10 lemons and they are asked to distribute them to the ideas they feel are less likely to win. From the participant's perspective this strategy looks and feels exactly like the BoL strategy. The difference is that after each participant submits their rating, the ideas are ranked by a greedy algorithm, which optimizes for both quality and diversity. Dynamic ranking is used, as in the other two strategies. Idea diversity is calculated using a submodular diversity function, which rewards idea difference (the more different an idea is to the ones already shown to the user, the higher reward it is given by the function). The metric we use to reward idea difference is inspired by the diversity reward function used by Lin *et al.* (2011) for multi-document summarization. This function rewards diversity and utility of a set S of items as follows:

$$f(S) = \sum_{j \in S} W_j + \lambda \times \sum_{c=1}^{K} \sqrt{|S \cap P_c|} \tag{1}$$

Here, $S = s_1, \ldots, s_m$ is a set of m items (in our case ideas, one idea in this set denoted by j). The more high-quality ideas and the more diverse ideas set S contains, the higher the value it is attributed by function $f(S)$. Set S is a subset of the original set V of all n ideas (i.e. $S \subseteq V$ where $V = v_1, \cdots, v_n$, and one idea in this set is denoted by i). The first part of the equation controls quality. W denotes the quality vector of the ideas in the set S at a given instance, such that a higher weight implies a better idea. Thus, the higher the quality of ideas set S contains, the higher the value of $f(S)$. The second part of the equation controls diversity. The set V of all ideas is partitioned

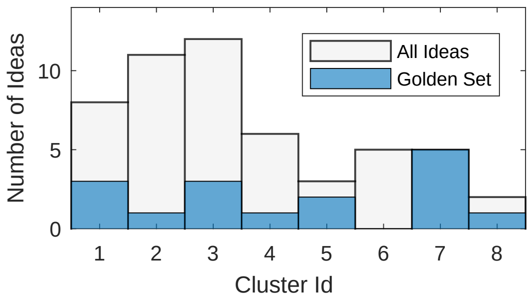


Fig. 2. Golden set versus total number of ideas

into k clusters. Each cluster P_c, c = 1,...,k contains a set of thematically similar ideas, and is disjoint from the rest of the clusters (i.e. $\bigcup_{c=1}^{K} P_c = V$ and $\bigcap_{c=1}^{K} P_c = \emptyset$). $|S \cap P_c|$ denotes the cardinality of the subset of S with ideas in cluster k. The square root function automatically promotes diversity by rewarding ideas from clusters that have not yet contributed to set S. Thus, the more ideas from underrepresented clusters that set S has, the higher the value of $f(S)$. Finally, the parameter λ controls the preference given to diversity over quality. A large λ value means that idea diversity will weigh more than quality.

Next, we use a submodular greedy algorithm (Algorithm 1) to order the ideas [36]. Given the set V of all ideas, the algorithm starts with an empty set S. In the end, this set S will be the ranking that the algorithm outputs. It will contain all ideas ordered in such a way as to maximize the objective value defined in Eq. 1, i.e. the ideas of high quality and high diversity (i.e. from clusters less represented so far) are at the top of the ranking. To achieve this, the algorithm starts adding ideas to set S and removing them from set V, one idea at a time, such that the selected idea $i \in V$ is the one with the highest marginal gain $\delta f(S \cup i)$ on set S. By choosing at each step to add the idea that will maximize quality and diversity of the existing set of already added ideas, the algorithm not only selects the ideas but also orders them as well. Finally, as the function in Eq. 1 is sub-modular and monotonic, the algorithm is also theoretically guaranteed to provide the best possible $(1 - \frac{1}{e})$ polynomial-time approximation to the optimal solution.

Algorithm 1: DBLemons algorithm. The algorithm performs a polynomial-time greedy maximization of the gain on the weighted combination between idea quality and diversity (Eq. 1). The output is a ranking of all ideas such that high-quality/high-diversity ideas are at the top.

```
   Data: Original set $V$ of all ideas
   Result: Ranked set $S$ of all ideas
1  initialization;
2  $S \leftarrow \emptyset$;
3  while $V \neq \emptyset$ do
4      Pick an item $V_i$ that maximizes $\delta f(S \cup i)$;
5      $S = S \cup \{V_i\}$;
6      $V = V - V_i$;
7  return $S$;
```

Calculating the λ value The DBLemons algorithm can function with all possible values of λ from Eq. 1. This means that it can be tuned to favor idea quality or diversity or none, depending on the needs of the ranking. A λ value equal to zero would mean that the algorithm is similar to BoL.

Since in this paper we examine the effect of diversity, we need to set the value of λ at a high enough value, so that diversity is favored in its output ranking. As an example, think of the following case. Suppose that in adding a new idea to the ranking the algorithm has to choose between (1) an idea with a high quality score from a cluster that has already been represented or (b) an idea with a lower quality score from a new cluster (emphasizing diversity). In such cases, we select a value of λ such that DBLemons will always favor diversity and use quality only as a tie-breaker between ideas of the same cluster

To calculate the λ value, which gives the desired diverse ranking, we proceeded as follows. We first take a random uniform distribution of lemons on ideas and then vary the value of λ using a step of 100 from 0 to 10^4 (as we will see, this upper value is more than enough to allow stabilization of the obtained result). We obtained similar results when using other distributions as well, namely the normal distribution, the log-normal distribution and the beta distribution parameterized in five different ways (altering the α and β shape parameters of the distribution, which allowed us to cover a very wide variety of possible vote distributions). Here, for brevity, we report results from the uniform and beta distributions. For each λ value we calculate the rank order of ideas. Then we compare the obtained ranking to the ranking that is acquired from the previous λ value. This provides a measure of how the ranking changes between consecutive values. To compare two different rankings, we use normalized Kendall's tau distance, a correlation metric widely used to compare ranking of items.

To avoid any outlines in the results due to the randomness of the distribution parameterization, we run the experiment 100 times per λ value and average the results. As the λ value increases, the obtained ranking changes, because diversity begins to have an effect on the ranking; however the ranking should stabilize when the λ value is sufficiently large. Since in this paper we examine the effect of diversity on crowd-based idea filtering, we are interested in finding the minimum value of λ after which diversity can fully produce its effects, i.e. the value of λ after which the obtained ranking stabilizes for maximum diversity. Fig. 3 illustrates the progress of the mean rank correlation with the progress of the λ value from 100 runs for six different distributions (uniform plus the five beta variations). We observe that for all values above $\lambda = 2000$, the obtained rankings are exactly the same. This gives us an empirical estimate of the minimum value of λ for diverse ranking for the dataset and lemon set size used in this paper. Selecting any value above the estimated cut-off will give the same results assuming the true vote distribution is similar to the considered distributions.

Rule of thumb for calculating the λ value. The minimum λ value described above requires a stepwise experiment, like the aforementioned, to identify. For the practitioner or researcher who wants to implement DBLemons inside their idea filtering system, running this experiment may not be feasible feasible, as they may not know the expected distribution of votes. Therefore, below we provide a "rule of thumb" approach for calculating the λ value. Note that this rule of thumb is based on the worst-case scenario, i.e. the extremely unlikely case that all voters give all their lemons to one single idea, and therefore produces a more conservative (i.e. higher) minimum value than the detailed experiment above, which, as we saw, will still produce the same ranking. Nonetheless, its advantage is that it can be easily adapted to different idea dataset, lemon bucket and voter population sizes. We proceed as follows. Our largest cluster has 14 ideas. Therefore the minimum marginal gain of adding a new element is $0.136(\sqrt{14} - \sqrt{13}$ in the square root function part of Eq. 1). In our experiments, we use 50 workers giving them 10 lemons each. This means that the maximum possible quality score for one idea can be 500. Hence, to ensure that diversity is always favored above quality, one can simply select any value of $\lambda > (500/0.136) = 3676$. As the true cut-off will always be lower than this value, the rank ordering will be exactly the same irrespective of which

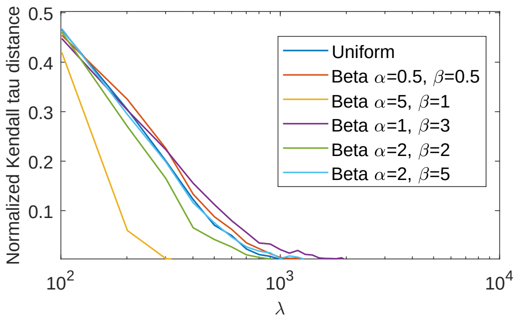


Fig. 3. Correlation of ranking between successive values of λ for DBLemons measured using Kendall tau distance (log scale for x axis). After $\lambda = 2000$ the ranking produced by DBLemons remains stable. This is the minimum cut-off value for the algorithm to show a clear preference for diversity.

value of λ is chosen above this cut-off value. In our experiments, we used a λ value well above the cut-off, namely $\lambda = 10000$.

Idea Clustering The greedy algorithm in Eq. 1 requires cluster labels of each idea for function evaluation. Often, these labels are provided by users themselves through tags when posting their idea to an open innovation contest. When this is not the case, one can obtain these labels either manually by placing similar ideas in buckets, or through automatic methods. In our study, we considered two methods of clustering the ideas: (1) automatic text-based and (2) manual concept-based. In the first method, we used the text of each idea to derive its word2vec vector representation, and calculate a similarity score between each pair of ideas (values in the 0 to 1 range). However, the clustering obtained using this method was unstable with ideas being allotted to different clusters in different clustering runs, mainly due to the fact that there existed little variation between the similarity of the different idea pairs (mean similarity of all ideas was found to be 0.85 and standard deviation equal to 0.04). Hence, we proceeded with a manual clustering of the ideas. We dig deeper into the effect of clustering, and discuss its impact on the performance for our task, as well as for future diversity-based methods, in the section 5.3.

For the manual clustering two experts, members of the research lab of the author team who had not seen the automatic clustering results, classified the ideas based on their thematic focus. They worked as follows. Each idea was printed in a physical card and spread randomly on a tabletop that served as the collaboration space. The experts then progressively grouped the ideas in thematic groups, moving the cards on the tabletop, through discussion and deliberation. Larger clusters appeared in the beginning, dividing the idea space into a few rough parts, and these were progressively refined to smaller clusters as the experts fine-grained the similarities and differences between the ideas. At the end of this process, eight clusters were identified, with ideas revolving around: 1) *Childcare facilities* (ideas around improving low-income women's opportunities through better childcare support), 2) *Education* (ideas focusing on improving the access to and the quality of training programs for women and girls), 3) *Employment* (ideas focusing on empowering women through novel ways of increasing their monthly income, like community-supported entrepreneurship), 4) *Gender-bias awareness* (ideas focusing on behavioral training of young boys and men on subjects related to inequality and gender-based violence), 5) *Leadership* , (ideas focusing on increasing the leadership potential of women from low-income areas) 6) *Physical Objects* (ideas involving a physical object to improve the safety of women, like water cleaning systems), 7) *Public Spaces* , (ideas around improving women's safety in public spaces) and 8) *Transportation* (ideas around improving safety within transportation means). Fig. 2 shows the number of ideas per cluster and the number of ideas in the golden set.

4 EXPERIMENTAL RESULTS

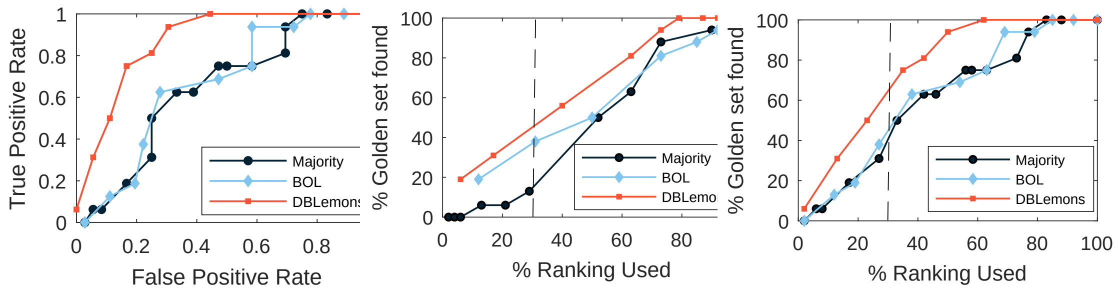
4.1 Experimental Setup

4.1.1 Testing platform. To test the three ranking methods, we developed a web platform with a look-and-feel similar to the OpenIDEO platform, with the difference that the ideas are displayed in the ranked order of the experimental strategy that is being used (Fig. 1). Ideas are displayed across multiple pages, with each page displaying 10 ideas. To see all 52 ideas, a worker has to go through 6 pages. However, replicating the OpenIDEO functionality, workers are free to evaluate as many ideas as they like, without the need to go through all of the ideas. Each idea is shown in a separate box that contains the idea's image, text summary and an evaluation option. In case majority voting is used, this evaluation option is a “thumbs up" button that the worker can activate (meaning that they upvote the idea), and workers can upvote as many ideas as they like. In case Bag of Lemons or Diversity is used, the evaluation option is a button that adds a number of “lemons" to the idea. Each worker starts with exactly 10 lemons, and they can allocate any number of these lemons to any idea. On the top right corner of the platform we placed a short instructions message, reminding the worker to vote for their preferred ideas (majority voting), or reminding them their number of unassigned lemons (BoL or DBLemons). Workers can change their evaluations (thumbs up or lemon assignments) as many times as they like, until they are satisfied with the result.

4.1.2 Crowd workers. We hired 150 Figure Eight workers (different than the ones used for the idea dataset creation described in the previous), and divided them randomly into the three experimental conditions (one condition for each strategy, 50 workers per condition). No specific test was required for workers to register for the Figure Eight task that gave access to our platform, since we aimed for the typical internet user, like the ones that usually vote in open innovation contests. Nevertheless, to ensure a minimum guarantee of task attention we opted for hiring mid-experienced but not over specialized Figure Eight workers (Level 2 out of 3). Each worker was given a link to the platform, and they were paid once they finished their evaluation. As noted above, similarly to real-world open innovation platforms, each worker was free to spend as much or as little time as they wanted reading and evaluating some or all of the ideas of the challenge. Nevertheless, to avoid bias in the results from possible spammers, we cleared the results of those workers that did not rate any idea or did not give any lemon (less than 2% of the hired workers) and replaced them with other workers to reach the desired number of hires (50 workers per compared strategy). Payment was calculated on the basis of $12/hour[6], and for an average estimated task duration of 15 minutes[7], i.e. $3 per worker. To further incentivize workers into making as qualitative evaluation as possible, we notified them that the Top 3 accurate raters would get an additional bonus of $1.

4.1.3 Ranked order calculation. As soon as all workers had finished voting, we obtained the ranked order for each of the three voting strategies. For the Majority voting case, ideas were ordered in descending number of total votes. For the BoL and DBLemons cases, ideas we ordered in ascending number of total lemons. In case of ties, we applied the dense ranking allocation model (“1223 ranking"), according to which when two ideas have the same number of upvotes or lemons, these ideas are given the same ranking place number (e.g. place number 2), and the idea(s) after them receive the next ordinal number (place number 3 in our example). To break ties within the same ranking place, we ordered ideas alphabetically.

[6]http://guidelines.wearedynamo.org
[7]As we will see in the results analysis, this time estimation indeed included the average task durations of all three algorithms.



(a) ROC curve, all 50 voters (b) Filtering efficiency, 20 first voters (c) Filtering efficiency, all 50 voters

Fig. 4. Performance comparison of the three ranking strategies. The dashed line shows the golden set cardinality cut-off. DBLemons finds more winning ideas, earlier on, with less workers, and using a smaller portion of the ranked idea space.

4.2 Performance

4.2.1 DBLemons outperforms BoL and Majority voting in filtering efficiency. Fig. 4 compares the filtering performance of the three strategies. We first show the final ROC curve, followed by filtering efficiency curves by using (b) 20 and (c) all 50 voters per strategy. From the ROC curve, it is evident that DBLemons outperforms both BoL and Majority voting with a higher AUC ($AUC_{Majority} = 0.648$, $AUC_{BoL} = 0.671$, $AUC_{DBLemons} = 0.869$). DBLemons also achieves a True Positive Rate (TPR) of 1, with False Positive Rate (FPR) of only 0.44. The other two methods perform similarly to one another, achieving their maximum TPR at FPR of 0.77 and 0.75.

Going a bit deeper into our analysis we observe Fig. 4 (b) and (c). Here, the y axis corresponds to the percentage of golden set ideas identified, using the x% first ideas of the ranked order of each strategy (x axis). For example in Fig. 4c, we see that considering all available voters and using the 50% first ideas of the returned ranked order, majority voting manages to retrieve approximately 60% of the golden set, BoL has exactly the same result, while DBLemons retrieves 94% of the golden set. The dashed vertical line corresponds to the cardinality of the golden set ($16/52 \approx 31\%$ of the total number of ideas). Two main observations can be made using this figure. First, the filtering efficiency of DBLemons is higher than both the BoL and Majority voting strategies, even from the first few voters (Fig. 4 b)). When all voters are used, DBLemons manages to identify three quarters (75%) of the golden set using approximately as many ideas as the golden set itself (35% of the ideas when the golden set represents 33% of the ideas). On the other hand, using the same percentage of ranked ideas, majority voting manages to find only 50% of the winning ideas. BoL is left even further behind as it starts making distinctions among ideas using 40% of its ranked order and above. More important, DBLemons manages to identify 94% of the winner ideas using only half (50%) of the idea space, while BoL needs to explore 70% and Majority voting 80% of the idea space respectively.

As we will see in the Discussion section 5 that will follow, the $20 - 30\%$ *reduction of the idea space size achieved by the proposed method compared to the other two alternatives, means significant gains in terms of effort and cost, and considerably improves the prospects of open innovation adoption by large commercial players.*

4.2.2 DBLemons has a higher distinction ability than the other two strategies. The second remark that we can make concerns the distinction ability of the three strategies. As it can be expected, the fewer people have voted, the more ties we have among the ranked ideas, and therefore the less distinction capability the algorithms have. This is more clearly reflected in Fig. 6, which visualizes the ranking changes per algorithm as the number of participating voters increases. Colored boxes in this figure represent golden set ideas, and blocks of boxes represent ideas that have received the same number of votes/lemons. For example, on the far left column, we see that using a limited (20) number of voters the Majority voting strategy distinguishes 11 total blocks of, i.e. ranks the 52 total ideas with 11 ranked places. To surpass the size of the golden set, and thus give a conclusive answer about ideas can be included in the Top 30% winning ideas, it needs a total of 26 ideas (blocks 1-7), out of the total 52. This means that for this specific number of voters, the distinction capacity of majority voting starts from 50% of the idea space. Coming back to Fig. 4 (b) we see this result quantitatively: indeed the point of the Majority voting line that surpasses the vertical 30% cutoff line starts at 50% of the x axis. Still on Fig. 4 (and confirming visually with Fig. 6) we can observe that *DBLemons can distinguish enough ideas to reach the golden set size earlier on than the other two algorithms.* For a few voters its distinction capacity starts at 40% of the idea space and surpasses the other two algorithms by 10%, while for 50 voters it requires only 35% of the idea space and is matched only by Majority voting.

4.2.3 BoL and Majority perform similarly in most cases. A last result in terms of performance concerns the comparison between BoL and Majority voting. We observe that when the 20 first voters are employed (Fig. 4b) and less than 60% of the ranked idea space is used, BoL performs better than Majority voting, finding 50% of the golden set. This result is consistent with [29], which also finds that Bag of Lemons performs better than Majority voting. However, as the percentage of the ranked idea space used by the strategies increases above 60%, the performance of BoL drops comparatively to Majority voting, and the two strategies perform similarly. This similar behavior becomes more apparent when more (=50) voters are employed (Fig. 4c), in which case the performance of the two strategies is very similar from the start. Although this result does not contradict with [29] due to the different baselines used, it opens up new questions on the applicability of BoL. BoL is reported to perform better in idea filtering when compared to Likert scale voting, while BoL with dynamic ranking is shown to perform similar to Majority voting in most cases of our experiments as seen above.

4.3 Time on task: BoL and DBLemons take equally less time than Majority voting

Although DBLemons provides better filtering accuracy, one may think that workers take more time in that method. However, we found that both DBLemons and BoL took 42% less median time compared to majority voting (Fig. 5). There are two possible explanations for the difference in timing. Prior literature [29] has argued that BoL uses less time as it only requires people to identify ideas that are clearly deficient with respect to at least one criterion. This can be a reason for the lesser time taken by both methods using lemons. However, we also observed that the total number of lemons allocated to the ideas is 43% of the total votes in majority voting. This may imply that workers took similar time per vote (or lemon allocation) in all three conditions. We tested this theory by observing time on task for workers using less lemons. One would expect that workers who allocate less lemons take proportionally less time. However, we did not find clear evidence of this from our dataset, as most workers used all ten lemons. We believe that the less time taken by the lemon-based methods can be attributed to a combination of both factors.

Summarizing, in answer to the two research questions of our related literature analysis, our results show that:

- Bag of Lemons with dynamic ranking (BoL) does not outperform Majority voting in terms of filtering efficiency; however it is speedier in finding good ideas, confirming the analysis by Klein and Garcia (2015).

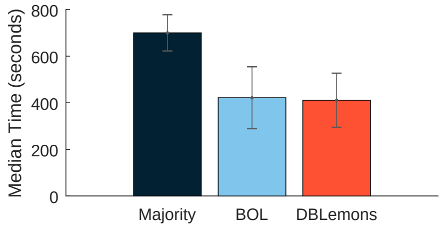


Fig. 5. Median task times

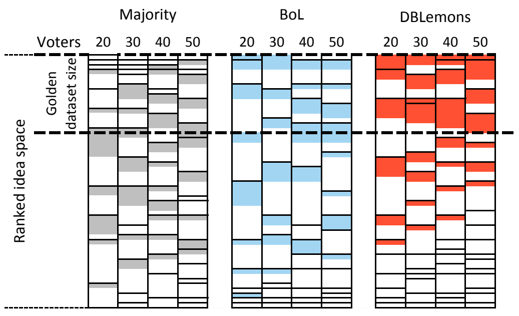


Fig. 6. Progressive ranking per strategy, descending quality order. All strategies improve with the number of voters, but DBLemons does so faster.

- Diversity-assisted Bag of Lemons (DBLemons) does increase the BoL efficiency in filtering high-quality ideas.

5 DISCUSSION

5.1 Impact on Open Innovation

Overall, our results give rise to two main observations, which impact open innovation:

- Majority voting, used widely by open innovation platforms, is more time consuming and less accurate than the other two alternatives.
- Diversification helps in the selection of high-quality ideas, more accurately than the other two alternatives and in less or equal time.

In specific, we found that BoL is speedier in finding good ideas than Majority voting (as also shown in [29]); however neither BoL nor Majority voting are very accurate. BoL is capable of detecting only 50% of the golden set ideas using 40% of the ranked ideas. Although, this is better than random chance, it is of less practical use. To achieve about 95% accuracy, BoL and Majority voting offer only a 20%-30% reduction of the idea space. In comparison, the DBLemons algorithm captures 94% of the winning ideas with 50% reduction of the idea space and 100% of the winning ideas with a 35% reduction.

5.1.1 ***Return-On-Investment.*** In practical terms, our results show that *with DBLemons a person has to look only at half of the actual ideas to get almost all of the winners*, and that DBLemons also saves this person 20% and 30% of evaluation effort compared to BoL and Majority voting respectively[8]. Assuming that a company wants its experts to evaluate an idea space that includes almost all (95%) of high-potential (golden set) ideas, these experts would have to go through 20% and 30% more ideas if the BoL and Majority strategies respectively were to be used. Considering: i) a median estimate of approximately 100K ideas received per large-firm ideation platform[9], ii) an average of 10% of these ideas reaching the latter stages of the innovation contest (as it is the case with OpenIdeo for example) and iii) an estimated expert cost of $500 and four hours to evaluate one idea in a Fortune 100 company [41], DBlemons can save organizations between $1 and $1.5M in costs and 50-75 person-months in effort. This constitutes a significant Return-on-Investment not

[8]Evaluation effort is measured in terms of the idea space percentage that a person has to go through after crowd-based filtering.

[9]From [29]: IBM's "Innovation Jam" gathered 46,000 ideas, Dell's IdeStorm 20,000 ideas, Google's 10 to the 100th project over 150,000 ideas, and Singapore Thinathon's contest over 454,000 ideas.

only for the proposed method, but most importantly for the prospect of open innovation in being considered as a viable complement to in-house innovation by large corporate players.

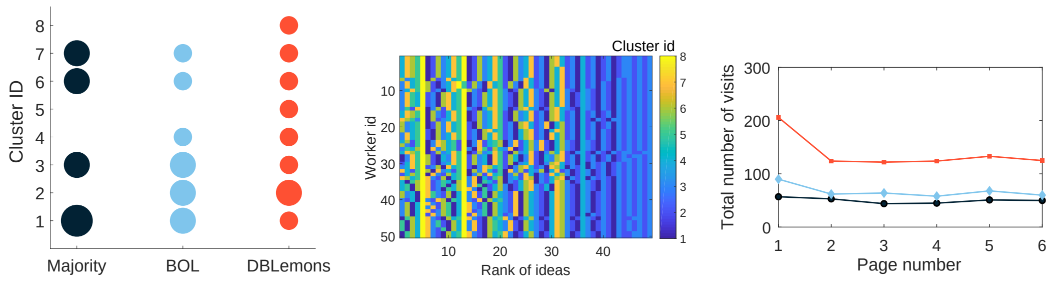
5.2 How diversity helps

A key question is why diverse ranking works so well, and under which circumstances it is expected to do so. We considered three possible explanations. First, it may be that the actual set of golden ideas have an equal representation across all clusters. By enforcing an equal number of good ideas from each cluster on the first page, DBLemons would increase the chance of getting a similar distribution. However, as it can be seen in Fig. 2 this is not the case; ideas are represented unequally and disproportionately to the cluster size, across the clusters. Hence, this potentially disadvantages DBLemons by pushing high quality ideas from same cluster down the ranking. Another reason can be that using lemons instead of upvotes helps in idea filtering. However, by comparing BoL with Majority voting, we saw that lemons by themselves are not more effective than upvotes, although they do help in reducing time on the voting task.

Finally, we argue that DBLemons works better as it provides better coverage of the idea space. We observe Fig. 7 (a) depicting the median idea clusters shown by the three algorithms on the first page. Each circle depicts the presence of an idea cluster on the page, and size is proportional to the idea percentage the cluster occupies. Whereas DBLemons always represents all 8 clusters, majority voting shows ideas from only half of the available clusters, while BoL omits two. A similar pattern is repeated across the rest of the pages. Focusing on the behavior of the DBLemons algorithm, Fig. 7 (b) illustrates the clusters seen by each worker when entering the platform and using DBLemons.

The y axis shows individual workers and it has 50 values, one for each worker. The x axis depicts the ranking of the 52 ideas seen by each worker. Every 10 ideas (values of the x axis) represent one consecutive page of the ranking. In other words, this figure contains 50 horizontal slices, with each slice representing the idea ranking seen by one individual worker. The color for each idea represents the cluster that the idea belongs, and we have 8 colors/clusters. This color coding allows us to observe that the left part of the figure has successive items with different colors. This means that all the workers saw ideas from different clusters at the start of their ranked list. We also observe that colors alternate significantly until approximately idea 34 (x axis value equal to 34). This means that for the first few pages (until page 4 out of 6) the algorithm represented all 8 clusters in equal proportions for all workers. This is indeed the expected behavior of DBLemons ranking, as observed in Figure 7a. Finally we observe that from the middle of page 4 (x axis value equal to 35) and until the end of the ranking (end of the x axis) the colors are similar. This reflects the fact that the ideas that belong to both a large cluster and have received many lemons (i.e. are of low quality), are correctly pushed by the algorithm towards the end of the list.

Essentially, DBLemons tries to maximize coverage over clusters, and in doing so it gives voters an overview of the full idea space right from the beginning. As people visit more pages, they realize that they have already seen similar ideas and can make faster and more informed comparisons, but also to go back and correct their previous evaluations. In contrast, people who see few concepts initially may get fixated on them. This observation is supported by literature on fixation, which proposes that the solution search process should begin with a divergent step prior to convergence [33]. Smith *et al.* (1993) referred to fixation as something that blocks or impedes the successful completion of cognitive operations (like remembering, solving problems and generating creative ideas), and proved that one's recent experience can lead to unintentional conformity or fixation to a few ideas. The fixation effect is further confirmed by looking at the workers' navigation behavior (Fig. 7 (c)). As we can observe, the volume of page visits for the DBLemons is approximately four times higher than that of Majority voting and two times higher than that BoL for the first page, while for the rest of the pages the DBLemons volume is twice as much than that of the other two strategies.



(a) Entropy of idea clusters - first page (b) DBLemons - Progressive rankings (c) Page visit volume

Fig. 7. DBLemons provides a more even coverage of all idea clusters: (a) Median cluster entropy, and (b) Cluster distribution for ranking seen by successive workers. This could lead voters to make more idea comparisons, generating (c) more page visits.

This higher volume, which in the dataset is due to multiple back and forth hops across the pages, supports the possibility that after getting a diverse summary of ideas from the first page, users made comparative decisions regarding their lemon allocation.

5.3 The effect of clustering

Our work examines the effect of diversification on crowd-based idea filtering. As such, we have maximized the coverage of clusters and obtained them through manual labeling, involving two collaborating experts, to minimize noise. Manual labeling can be also obtained through the idea authors, who often categorize their proposed ideas upon submitting them to an open innovation contest (like in the case of our generalization experiment with the UNESCO dataset), or it can be crowdsourced to multiple independent evaluators (in which case elements like inter-coder reliability need also to be examined). However, one may wonder what happens in case the cluster labels are noisy, an issue frequently encountered when fully automatic clustering is employed. To check this behavior, we ran an additional experiment using automated clustering.

We represented each idea as a vector, using Google's publicly available pre-trained word embeddings[10], and then summing these to obtain the sentence embeddings of the idea. This is a widely used method [22], which has also been adopted by the Semantic Textual Similarity shared task [10]. Next we used cosine similarity to compute the similarity between all pairs of ideas, a process which revealed that automatic similarity calculation could not differentiate much between the ideas (mean similarity 0.85, standard deviation 0.04, in the 0 to 1 range). On the obtained similarity matrix we applied spectral clustering with 8 clusters (same as manual clustering) to find the cluster labels for each idea. In doing so, we noticed that different spectral clustering runs provided different clustering assignments, i.e. that the dataset did not have well-defined automatically-identifiable clusters. To further ascertain this, we ran 100 clustering runs and calculated the silhouette score for each run. This score measures how well-defined the identified clusters are and is widely used in literature [6] to judge different clusterings. Silhouette coefficients near a value equal to +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters, and negative values indicate that those samples might have been assigned to the wrong clusters. The obtained scores varied between 0 to 0.1, showing poor clustering on the dataset. We nonetheless proceeded

[10]https://code.google.com/archive/p/word2vec/, providing 300 dimensional vectors for 3 million word embeddings, pre-trained on a Google News corpus of about 100 billion words.

with our further analysis, selecting the clustering assignment with the maximum silhouette score. Interestingly, this assignment also had the most – albeit still quite low – similarity with the manual labeling (maximum adjusted rand score 0.29), compared to the other assignments.

Using this automatic clustering assignment, we replicated the DBLemons experiment with 50 workers. We found that this method required 60% of the idea space to find 81% of the winning ideas (ROC $AUC_{DBLemons-auto} = 0.741$). This result is poorer than the original DBLemons, which only required 45% of the idea space for finding the same number of top ideas ($AUC_{DBLemons-manual} = 0.869$), but it is still slightly better than the BoL and Majority voting methods, which required 70% ($AUC_{BoL} = 0.671$) and 80% ($AUC_{Majority} = 0.648$) of the idea space respectively.

To cover the case that this result is due to the particular automatic clustering method used, we also compared four additional clustering algorithms— KMeans, Gaussian mixture models, Ward Agglomerative and Affinity Propagation clustering. For each method, we conducted 100 runs and calculated the maximum silhouette score. Adding these algorithms allowed us to also experiment with fixed defined (8 clusters for the KMeans, Gaussian mixture, and Ward Agglomerative methods) versus non-fixed number of clusters (in Affinity Propagation method). Results with all these algorithms showed that automatic clustering, on the particular dataset, gives noisy cluster assignments with large differences across runs. The maximum silhouette scores (from 100 runs per algorithm) for these methods were 0.06, 0.04, 0.11 and 0.03 respectively. The values near 0 show that none of the methods was able to give good/decisive results from run to run in regards to the clustering, and none of the methods had significant advantage compared to the others. We also compared all methods (spectral clustering and the additional four above) using a second cluster fitness metric, namely the Dunn score. In this case too, no major difference or advantage was observed in the clustering capability of the algorithms. This noisy clustering result, across different clustering methods, can be attributed to the small size of text per idea, which does not allow the creation of a representative context corpus. In case the algorithms could be provided with context knowledge or in case the text per idea was longer, automatic clustering may have given better results.

Concluding, the relatively poor performance of automated clustering compared to manual clustering was not surprising, since the automated method did not differentiate much between ideas and the idea vector space lacked well-defined clusters. This noise meant that automatic clustering risked placing ideas that are very different from one another into the same cluster, a fact which we also manually verified (see for example Table 1). Hence, we observe that the effectiveness of DBLemons partially depends on idea clustering. We believe that this is true for any method that tries to leverage the power of diversity. If noisy labels are assigned to the ideas, any method maximizing coverage over these labels will also be affected, therefore a practitioner must take special care in preparing the idea clusters or defining their similarity. Nevertheless, the fact that even with noisy input DBLemons still manages to obtain better results than the compared alternatives, indicates that the potential of diversity when combined with crowd evaluations is important, and will continue improving as better clustering methods appear by independent research.

5.4 Generalization of the results

Another important topic for discussion is the generalizability of the proposed approach. The results elaborated so far are based on the dataset derived from OpenIdeo, because OpenIdeo is among the leading platforms for crowd ideation. For completeness purposes however, it is important to test the proposed approach on another ideation dataset, preferably one from a different platform and challenge context, and one subjected to as little processing as possible in regards to the creation of the golden dataset, the clustering of the ideas, or the idea text creation. The dataset we worked with comes from the 2017 Youth Citizen Entrepreneurship competition of UNESCO's Global Action

Idea 1	**Idea 2**
This idea proposes a participatory planning process, which enables communities to design safer public spaces for women through safety audits. Safety auditing is the process of gathering data about the safety of a place, and it is usually based on data gathered through smart phones. Unfortunately smart phone usage is limited in countries like India. To solve this issue, local community members and volunteers conduct safety audits of public spaces and data is collected through not only mobile, but also through online or face-to-face meetings. The analyzed data is displayed on a large interactive map in an open public space such as a park. This map can then be used as a canvas on which local community members can write and draw their reactions and suggestions to the safety audit information provided. The idea will be evaluated by tracking change in safety parameters over time.	This idea proposes a platform that enables women to work from home. The platform consists of a network of income generation modules, providing women with appropriate equipment to work, especially in poor housing areas. These physical modules can be easily attached to low-income houses, with properly designed working spaces and online connectivity. Women can choose their preferred module based on local skills and demands. The platform will help reduce the time of traveling to work and directly improve women's safety, while also giving them more time for child care. Continually educating women will also help them understand their work rights and the benefits of not including children in work. The idea will be implemented gradually, from system design and mapping the viable income generation alternatives, to prototyping and launching the platform. The evaluation will be made through qualitative and quantitative data that assesses the platform's impact on empowering women.

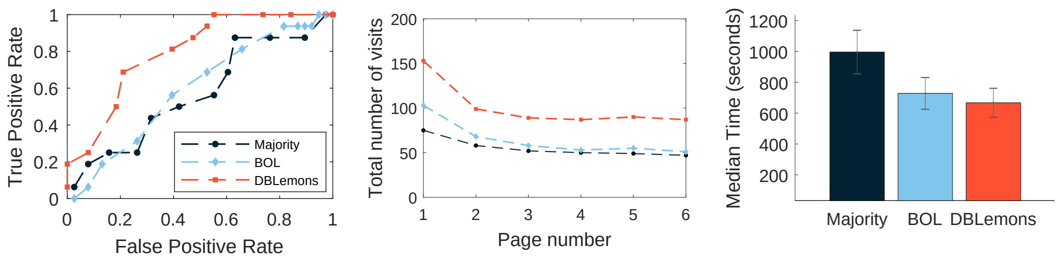
Table 1. Two conceptually different ideas incorrectly clustered together by automated clustering. Manual clustering assigned them to different clusters: the first idea to "Public Spaces" and the second idea to "Employment"

Programme on Education for Sustainable Development[11]. The competition called for innovative ideas and projects to address important social, economic, environmental, health and governance challenges of our times. The 176 ideas of the dataset were already categorized into one or more of 17 thematic clusters, according to the Sustainable Development Goal that they work towards solving. We used the same clusters, in order to avoid any potential bias from the manual clustering approach used for the original OpenIdeo dataset. In case an idea belonged to two clusters, we used the cluster marked by the idea authors as primary. We also did not intervene in the creation of the golden dataset, which was taken directly from the competition data based on the number of comments per idea. Using the number of comments as a quality indicator was supported by the fact that the ideas with the highest number of comments were also those nominated by the competition panel of judges as the winners. Finally, to avoid any potential writing style bias from summarizing the ideas, the text that we used per idea was the original "Explain your idea in details" paragraph included in each idea description.

The UNESCO challenge, similarly to the OpenIdeo challenge, received ideas of various quality levels: from excellent, to good, to mediocre, and finally to incomplete ones. To maintain comparability with our original results, but also taking into account that the focus of this paper is using crowd filtering to sort the good ideas from the excellent ones, from the 176 ideas of the original dataset, we worked with the first 54 ideas in descending quality order (selecting 52 ideas, to match precisely the size of the OpenIdeo dataset was not possible, as the last four ideas of the UNESCO dataset had received the exact same number of comments, so we included them all). The number of clusters these ideas belonged to was 10, which again is comparable to the number of clusters of the original OpenIdeo dataset (8 clusters). The golden dataset consisted of the 16 most high-quality ideas, equal to the golden dataset size of the original experiment. Finally, similarly to the original experiment, we hired 50 Figure Eight crowd workers per experimental condition/algorithm.

Results, illustrated in Fig. 8, showed that the three compared algorithms performed similarly in relation one to the other as in the main experiment. In specific, the algorithms exhibited similar behavior in terms of performance (ROC curves, compare Fig. 8a and Fig. 4a), page visit volume

[11]https://www.entrepreneurship-campus.org/ideas/12/



(a) ROC curves (b) Page visit volume (c) Median task times

Fig. 8. Examining the generalization of the proposed approach on a different dataset (UNESCO's Youth Entrepreneurship ideation contest). The three algorithms exhibit a similar behavior in comparison to one another as in the main experiment.

(compare Fig. 8b and Fig. 7c) and median task time (compare Fig. 8c and Fig. 5). All three algorithms demonstrated a small drop in performance compared to the OpenIdeo experiment, and such a variation can be expected since the two datasets refer to different idea competitions and context. Observing this similarity in performance between the two datasets (OpenIdeo and UNESCO) reinforces the generalization potential of the proposed approach, for the specific stage of the innovation process (filtering the excellent from the good crowd ideas) and dataset size. The topic of generalization however is very broad. Different contexts, e.g. those involving ideas from experts (rather than from the crowd), or innovation stages (e.g. larger datasets from the initial stages of an idea contest that include a lot of incomplete or stub ideas) may affect generalization. Exploring these parameters can be the topic of further experiments and future research.

5.5 Limitations and Future Work

5.5.1 Scope. In this paper, we focused on the third (out of the four) stages of an OpenIDEO challenge. As such, our work can claim generalization only for the latter stages of open innovation, when the very low-quality or stub ideas have already been filtered out and it is harder to distinguish the good from the excellent ideas that will be retained for elaboration and funding. For a fully streamlined end-to-end crowd filtering process, which can return to the experts a minimum number of ideas and yet still contain almost all top-quality ones, in the future we aim to examine our approach on earlier stages of the open innovation process. Alternatively, it would be interesting to combine our existing approach with reference-based scoring models (e.g. [51]) that filter out large idea sets, by restricting crowd voter access to only a few representative ones. This would combine the advantages of both a quick filtering for the large mass of initial ideas, and of a more fine-grained one performed by our method for the latter stages. In the future, we would also like to examine this combination.

5.5.2 Algorithm and setting variations. In the version of the algorithm used in this paper, and for the DBLemons and BoL strategies, we gave voters a fixed budget of 10 lemons, corresponding to 19% of the idea corpus. This choice was made to be able to compare, to the best possible extent, our results with those of the original BoL strategy [28, 29]. However, changing the number of lemons is expected to affect (increase or decrease) the expressive power of voting. Consider for example the extreme case each user has only one lemon. It is likely that they will try to allocate this lemon to one of the worst ideas, but it is also likely that the ranking will be unable to distinguish between the rest of the ideas because of vote sparsity. On the other extreme, allocating a very large number of lemons will be time consuming for the users, and it may affect the time of the task and possibly their accuracy. In the future we would like to systematically vary the number of lemons to study the effect of choice number on the filtering efficiency and on the time on the task. On a related remark, majority voting was implemented in this paper as a single-voting strategy (where a user can allocate up to one vote per idea). This choice was motivated by the way majority voting is usually implemented in open innovation platforms (like OpenIdeo), and in order to be able to construct and compare with a realistic benchmark. In the future, it would also be interesting to examine a multi-voting adaptation of this strategy.

Furthermore, in this work, voters did not see the ratings provided by others (although they are indirectly exposed to them due to the dynamic ranking applied on all three strategies). A future research direction could thus be to investigate the impact of information cascades on the open innovation crowd filtering problem. Moreover, the version of DBLemons used in this paper gives equal weight to all clusters irrespective of their size. In the future it would be interesting to test alternative definitions of diversity, which give proportional weight to each cluster based on cardinality. We expect that this will mean that larger clusters will get more ideas at the top of the list. Another future research direction is to identify the minimum number of workers required for efficient DBLemons ranking, depending on idea complexity, cluster size and other variables. Finally, in this paper we are interested in exploring the effect of diversity compared to the previously used methods of BoL and Majority voting. Therefore for the implementation of DBLemons we chose a λ value well above the cut-off limit of section 3.2.3, which gives the algorithm a clear preference for diversity. In the future it would be interesting to explore hybrid rankings, where a less clear preference on diversity is given to DBLemons, using λ values between zero and the cut-off limit. Below this limit, the lower the value the more the algorithm will resemble BoL and the higher the value the more the algorithm will resemble DBLemons as it is described in this paper. We note that the subject of how much one should diversify the ranking is a persistent question in the domain of recommender systems and also a domain- and context-specific problem.

5.5.3 Interface Design for eliciting idea preferences. In open innovation contests, people are given access to the full set of candidate ideas [1]. Since this work has been about improving open innovation, we used the same type of interface, giving crowd voters the possibility to browse through all of the ideas if they wished to. However, there is also work [42] on eliciting preferences by prompting participants with comparisons that are dynamically chosen to ensure coverage and optimize for speed. For instance, [14] studied usefulness of social choice functions in crowdsourcing for participatory democracies and provided algorithms which efficiently elicit responses. In the future, it will be worth exploring the effect of these methods of preference elicitation for open innovation filtering.

6 CONCLUSION

In this paper we propose DBLemons, a new strategy for crowd-based idea filtering that combines the Bag of Lemons approach with a diversification of the idea concept space. Working with a dataset from an open innovation contest on women's safety, we show that DBLemons increases filtering efficiency and takes less time compared to majority voting (a popular open innovation filtering strategy), while compared to Bag of Lemons without diversity it also exhibits higher filtering efficiency. We attribute this to the larger number of idea comparisons made by voters in lesser time; this is possible since DBLemons shows representative ideas of all concepts early on. Overall, our proposed method contributes to improving trust in crowd-based idea filtering and hence it can help increase the strategic value of open innovation as an organizational governance choice.

7 ACKNOWLEDGEMENTS

The work of Dr. Konstantinos Papangelis and Mr. Irwyn Sadien has been supported by the Xi’an Jiaotong - Liverpool University grant: RDF 15-02-17, and by the Research Institute of Big Data Analytics of Xi’an Jiaotong-Liverpool University.

REFERENCES

[1] Simon à Campo, Vasssilis-Javed Khan, Konstantinos Papangelis, and Panos Markopoulos. 2018. Community heuristics for user interface evaluation of crowdsourcing platforms. *Future Generation Computer Systems* (2018). https://doi.org/10.1016/j.future.2018.02.028

[2] Faez Ahmed and Mark Fuge. 2017. Capturing Winning Ideas in Online Design Communities. In *20th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, Portland, USA.

[3] Faez Ahmed and Mark Fuge. 2018. Ranking ideas for diversity and quality. *Journal of Mechanical Design* 140, 1 (2018), 011101.

[4] Faez Ahmed, Mark Fuge, and Lev D. Gorbunov. 2016. Discovering diverse, high quality design ideas from a large corpus. In *ASME International Design Engineering Technical Conferences*. ASME, Charlotte, USA. https://doi.org/10.1115/DETC2016-59926

[5] Kamal Ali and Wijnand van Stam. 2004. TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’04)*. ACM, New York, NY, USA, 394–401. https://doi.org/10.1145/1014052.1014097

[6] Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Inigo Perona. 2013. An extensive comparative study of cluster validity indices. *Pattern Recognition* 46, 1 (2013), 243–256.

[7] Ivo Blohm, Christoph Riedl, Johann FÃijller, and Jan Marco Leimeister. 2016. Rate or Trade? Identifying Winning Ideas in Open Idea Sourcing. *Information Systems Research* 27, 1 (2016), 27–48. https://doi.org/10.1287/isre.2015.0605 arXiv:https://doi.org/10.1287/isre.2015.0605

[8] David V Budescu and Eva Chen. 2014. Identifying expertise to extract the wisdom of crowds. *Management Science* 61, 2 (2014), 267–280.

[9] Alex Burnap, Yi Ren, Richard Gerth, Giannis Papazoglou, Richard Gonzalez, and Panos Y Papalambros. 2015. When crowdsourcing fails: A study of expertise on crowdsourced design evaluation. *Journal of Mechanical Design* 137, 3 (2015), 031101.

[10] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation. *ArXiv e-prints* (July 2017). arXiv:cs.CL/1708.00055

[11] Liang Chen and De Liu. 2012. *Comparing strategies for winning expert-rated and crowd-rated crowdsourcing contests: First findings*. Vol. 1. 97–107.

[12] Liang Chen, Pei Xu, and De Liu. 2016. Experts versus the crowd: a comparison of selection mechanisms in crowdsourcing contests. (2016). https://doi.org/10.2139/ssrn.2542857

[13] Clyde H Coombs and George S Avrunin. 1977. Single-peaked functions and the theory of preference. *Psychological review* 84, 2 (1977), 216.

[14] Tanja Aitamurto Helene Landemore David Timothy Lee, Ashish Goel. 2014. Crowdsourcing for Participatory Democracies: Efficient Elicitation of Social Choice Functions. *Second AAAI Conference on Human Computation and Crowdsourcing* (2014).

[15] Douglas L Dean, Jillian M Hender, Thomas L Rodgers, and Eric L Santanen. 2006. Identifying Quality, Novel, and Creative Ideas: Constructs and Scales for Idea Evaluation. *Journal of the Association for Information Systems* 7, 10 (2006), 646–698. https://doi.org/Article arXiv:http://ehis.ebscohost.com/

[16] Shristi Drolia, Shrey Rupani, Pooja Agarwal, and Abheejeet Singh. 2017. Automated Essay Rater using Natural Language Processing. *International Journal of Computer Applications* 163, 10 (Apr 2017), 44–46. https://doi.org/10.5120/ijca2017913766

[17] Nadine Escoffier and Bill McKelvey. 2015. The wisdom of crowds in the movie industry: Towards new solutions to reduce uncertainties. *International Journal of Arts Management* 17, 2 (2015), 52.

[18] David Fisher, Ashish Jain, Mostafa Keikha, WB Croft, and Nedim Lipka. 2015. *Evaluating Ranking Diversity and Summarization in Microblogs using Hashtags.* Technical Report. Technical report, University of Massachusetts.

[19] Mark Fuge, Kevin Tee, Alice Agogino, and Nathan Maton. 2014. Analysis of Collaborative Design Networks: A Case Study of OpenIDEO. *Journal of Computing and Information Science in Engineering* 14, 2 (March 2014), 021009+. https://doi.org/10.1115/1.4026510

[20] Lidia Gryszkiewicz, Ioanna Lykourentzou, and Tuukka Toivonen. 2016. Innovation labs : leveraging openness for radical innovation? *Journal of Innovation Management* 4, 4 (2016), 68–97. https://doi.org/10.2139/ssrn.2556692

[21] Joel West Henry Chesbrough, Wim Vanhaverbeke. 2006. *Open Innovation: Researching a New Paradigm.* Oxford University Press.

[22] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 1367–1377. https://doi.org/10.18653/v1/N16-1162

[23] Tom Hope, Joel Chan, Aniket Kittur, and Dafna Shahaf. 2017. Accelerating Innovation Through Analogy Mining. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. ACM, New York, NY, USA, 235–243. https://doi.org/10.1145/3097983.3098038

[24] Eelko K.R.E. Huizingh. 2011. Open innovation: State of the art and future perspectives. *Technovation* 31, 1 (2011), 2 – 9. https://doi.org/10.1016/j.technovation.2010.10.002 Open Innovation - ISPIM Selected Papers.

[25] Marcus Matthias Keupp and Oliver Gassmann. 2009. Determinants and archetype users of open innovation. *R&D Management* 39, 4 (2009), 331–341. https://doi.org/10.1111/j.1467-9310.2009.00563.x

[26] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The Future of Crowd Work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 1301–1318. https://doi.org/10.1145/2441776.2441923

[27] Mark Klein and Gregorio Convertino. 2015. A roadmap for open innovation systems. *Journal of Social Media for Organizations* 2, 1 (2015), 1.

[28] Mark Klein and Ana Cristina Bicharra Garcia. 2014. The bag of stars: High-speed idea filtering for open innovation. (2014). https://doi.org/10.2139/ssrn.2387180

[29] Mark Klein and Ana Cristina Bicharra Garcia. 2015. High-speed idea filtering with the bag of lemons. *Decision Support Systems* 78 (2015), 39–50.

[30] Ulrich Lichtenthaler and Eckhard Lichtenthaler. 2009. A Capability-Based Framework for Open Innovation: Complementing Absorptive Capacity. *Journal of Management Studies* 46, 8 (2009), 1315–1338. https://doi.org/10.1111/j.1467-6486.2009.00854.x

[31] Hui Lin and Jeff Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 510–520. http://dl.acm.org/citation.cfm?id=2002472.2002537

[32] Christian List. 2012. Collective Wisdom: Lessons From the Theory of Judgment Aggregation. In *Collective Wisdom: Principles and Mechanisms*, Helene Landemore and Jon Elster (Eds.). Cambridge University Press.

[33] Y-C Liu, A Chakrabarti, and T Bligh. 2003. Towards an âĂŸidealâĂŹapproach for concept generation. *Design Studies* 24, 4 (2003), 341–355.

[34] A Majchrzak and A Malhotra. 2013. Viewpoint: Towards an information systems perspective and research agenda on crowdsourcing for innovation. *Journal of Strategic Information Systems* 22 (2013), 257–268.

[35] William L. Moore. 2004. A cross-validity comparison of rating-based and choice-based conjoint analysis models. *International Journal of Research in Marketing* 21, 3 (2004), 299 – 312. https://doi.org/10.1016/j.ijresmar.2004.01.002

[36] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functionsâĂŤI. *Mathematical Programming* 14, 1 (1978), 265–294.

[37] Michelle A Pang and Carolyn C Seepersad. 2016. Crowdsourcing the Evaluation of Design Concepts With Empathic Priming. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, V007T06A004–V007T06A004.

[38] GP Patil and Charles Taillie. 1982. Diversity as a concept and its measurement. *Journal of the American statistical Association* 77, 379 (1982), 548–561.

[39] John W. Payne, James R. Bettman, Eloise Coupey, and Eric J. Johnson. 1992. A constructive process view of decision making: Multiple strategies in judgment and choice. *Acta Psychologica* 80, 1 (1992), 107 – 141. https://doi.org/10.1016/0001-6918(92)90043-D

[40] Christoph Riedl, Ivo Blohm, Jan Marco Leimeister, and Helmut Krcmar. 2013. The Effect of Rating Scales on Decision Quality and User Attitudes in Online Innovation Communities. *International Journal of Electronic Commerce* 17, 3 (2013), 7–36. https://doi.org/10.2753/JEC1086-4415170301 arXiv:http://www.tandfonline.com/doi/pdf/10.2753/JEC1086-4415170301

[41] Alan G Robinson and Dean M Schroeder. 2004. *Ideas are free: How the idea revolution is liberating people and transforming organizations.* Berrett-Koehler Publishers.

[42] Matthew J. Salganik and Karen E. C. Levy. 2015. Wiki Surveys: Open and Quantifiable Social Data Collection. *PLOS ONE* 10, 5 (05 2015), 1–17. https://doi.org/10.1371/journal.pone.0123483

[43] Juho Salminen. 2015. *The role of collective intelligence in crowdsourcing innovation.* Ph.D. Dissertation. Lappeenranta University of Technology. http://urn.fi/URN:ISBN:978-952-265-876-0

[44] Steven M Smith, Thomas B Ward, and Jay S Schumacher. 1993. Constraining effects of examples in a creative generation task. *Memory & cognition* 21, 6 (1993), 837–845.

[45] James Surowiecki. 2004. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations.* Doubleday.

[46] John Sweller. 1988. Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science* 12, 2 (1988), 257–285. https://doi.org/10.1207/s15516709cog1202_4

[47] Eliot van Buskirk. 2010. Google struggles to give away 10 million. http://www.wired.com/2010/06/google-struggles-to-give-away-10-million/all/1. (2010). Archived on: 2016-02-04.

[48] Jeroen J. G. van Merrienboer, Paul A. Kirschner, and Liesbeth Kester. 2003. Taking the Load Off a Learnerś Mind: Instructional Design for Complex Learning. *Educational Psychologist* 38, 1 (2003), 5–13. https://doi.org/10.1207/S15326985EP3801_2

[49] Saúl Vargas and Pablo Castells. 2011. Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11).* ACM, New York, NY, USA, 109–116. https://doi.org/10.1145/2043932.2043955

[50] Thomas Wagenknecht, Jan Crommelinck, Timm Teubner, and Christof Weinhardt. 2017. When Life Gives You Lemons: How rating scales affect user activity and frustration in collaborative evaluation processes. In *13th International Conference on Wirtschaftsinformatik.*

[51] Anbang Xu and Brian Bailey. 2012. A Reference-based Scoring Model for Increasing the Findability of Promising Ideas in Innovation Pipelines. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12).* ACM, New York, NY, USA, 1183–1186. https://doi.org/10.1145/2145204.2145380

[52] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: Introducing Serendipity into Music Recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12).* ACM, New York, NY, USA, 13–22. https://doi.org/10.1145/2124295.2124300

[53] Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. 2007. Improving Diversity in Ranking using Absorbing Random Walks.. In *HLT-NAACL.* Citeseer, 97–104.