Can AI Help in Crowdsourcing?
A Theory-Based Model for Idea Screening in Crowdsourcing Contests

Abstract

Crowdsourcing generates thousands of ideas. The selection of best ideas is costly because of the limited number, objectivity, and attention of experts. Using a dataset of 21 crowdsourcing contests that include 4191 ideas, we test how AI can assist experts in screening ideas. The authors have three major findings. First, while even the best previously published theory-based models cannot mimic human experts in choosing the best ideas, a simple model using LASSO can efficiently screen out ideas considered bad by experts. In an additional 22nd hold-out contest with internal and external experts, the simple model does better than external experts in predicting the ideas selected by internal experts. Second, the authors develop an Idea Screening Efficiency curve that trades off the False Negative Rate against the total ideas screened. Managers can choose the desired point on this curve given their loss function. The best model specification can screen out 44% of ideas, while sacrificing only 14% of good ideas. Alternatively, for those unwilling to lose any winners, a novel two-step approach screens out 21% of ideas without sacrificing a single 1st-place-winner. Third, a new predictor, Word Atypicality, is simple and efficient in screening. Theoretically, this predictor screens out atypical ideas and keeps inclusive and rich ideas.

Keywords: Creativity, Ideation, Crowdsourcing, Prototypicality, Word Atypicality, LASSO, Random Forest, RuleFit, AI, Natural Language Processing.

1. Introduction

Artificial intelligence (AI) faces exciting prospects. It is transforming existing business tasks to be done faster, cheaper, and with higher quality. Managers consider AI to be the most important general-purpose technology of our times (Brynjolfsson and McAfee 2017). AI has the potential to change industries just like the internet did 30 years ago or like electricity did 100 years ago (Füller et al. 2022). In innovation, AI challenges what has been taken for granted (Cockburn et al. 2019). Currently, innovation managers see the huge potential of AI-assisted methods (Füller et al. 2022) but are uncertain how it can help in ideation and idea screening.

Idea generation and screening are fundamental to marketing success because they are the start of a new product (Toubia and Flores 2007). They belong to the "fuzzy front end," a key point of leverage in the strategy of the firm (Dahan and Hauser 2001; Eling, Griffin, and Langerak 2014). Crowdsourcing taps diverse information sources and generates a high volume of ideas at low-cost (Terwiesch and Ulrich 2009). Commercial crowdsourcing platforms offer ideation-related services (Luo and Toubia 2015).

The large group of ideas includes many redundant or poor ideas. Thus, crowdsourcing presents a new challenge in ideation: screening ideas to identify the best. This screening process can be performed by users, i.e., contest participants, Amazon Mechanical Turk Workers, or experts. Relying solely on contest participants for screening could be problematic because they may act strategically. For example, participants may vote down good ideas that compete with their own. Using Amazon Mechanical Turk Workers can be problematic because of their potentially low expertise for specialized contests. The more 'cutting edge' a product is, the more likely experts will be required (O'Quin and Besemer 1999). The use of experts is very costly because of their limited number, cognitive capability (Toubia 2006), attention span, or objectivity. A solution to this problem may be to let many experts each screen few ideas. Such work division may be effective (Toubia and Flores 2007). However, this approach may still be insufficient if the number of ideas is very large, which is common in crowdsourcing.

Any idea judged to be high quality by experts is taken to the next stage of development. Costs increase exponentially as ideas progress through the new product funnel, from generation and screening,

to development, prototyping, market testing, and commercialization. Errors in screening can end up being very costly for firms. Urban and Katz (1983) gave the following justification for the cost of ASSESSOR, a tool they created for screening new products: "[it is] a screening device intended to eliminate product failures at a low cost (for example, $50,000) rather than carrying them on to test market where they would be rejected at a high cost (for example, $1-2M)." Thus, idea screening is critical to reducing large future costs of new products.

Besides the limited number of experts, their capacity to judge is also limited. When faced with many ideas, tedium sets in, and the judgement of experts may fluctuate or deteriorate. In this context, idea screening can be valuable as it allows experts to focus on a smaller number of the best ideas. Any mechanism used to screen ideas will carry Type I and II errors. A Type I error means wrongly selecting a bad idea (potential loser) while a Type II error means wrongly screening out a good idea (potential winner). In companies, managers have no choice but to accept some errors. In the words of Urban and Katz (1983) "the manager's task, therefore, is to set GO/NO cut-off values that balance these errors and maximize the firm's expected profit." This implies that tools which can aid in the setting of such cut-off values are of substantial value to managers.

AI models may aid in screening at the idea stage of the new product development process, where the new product funnel is widest. AI models (Goodfellow, Bengio and Courville 2016) have several advantages over human experts. First, once developed, AI models are relatively low cost to operate. Second, they do not share internal biases or succumb to adverse incentives. Third, they are private, so firms can use them as decision aids without disclosing sensitive intellectual property to third parties. Fourth, they do not tire. Fifth, theory-based AI models are transparent and not black boxes.

An entirely different approach is to use multi-armed bandits (e.g., Jain, Mason, and Nowak 2017; Jamieson et al. 2015; Katariya et al. 2018; Sievert et al. 2017). This study is an early attempt in AI to evaluate ideas based on theory-based models. Theory allows for generalizability of findings to other datasets because it provides an understanding of *why* ideas are good or not. The models tested also have low data requirements: most need only the text of the ideas to work. In sum, the theory-based models

have at least four advantages over bandits: they are instantaneous, generalizable, low cost, and suffer no confidentiality concerns because managers and researchers do not have to show the ideas to outside experts for judgments. The last advantage is important, for example when companies search for innovations of high confidentiality and/or high strategic performance.

Screening is the first of three levels where AI could help in ideation. Ideators would still be needed for idea generation and experts for idea selection. The second level is selecting the best ideas, thus bypassing experts altogether. Ideators would still be needed for idea generation, but machines could replace humans for idea selection. The third level is generating best ideas. This third level, if automated, would eliminate the need for ideators and make crowdsourcing obsolete.

The data for our study come from Hyve, an innovation company, which runs a crowdsourcing platform for idea generation and selection. We asked the crowdsourcing platform's director to specify a threshold of accuracy that would satisfy Hyve's clients, i.e., for a useful cost function. He gave us two thresholds of accuracy: screen out 25% of all ideas without sacrificing more than 15% of good ideas or screen out 50% of all ideas without sacrificing more than 30% of good ideas. We use Hyve's two criteria to construct a reference line we then compare to our proposed Idea Screening Efficiency (ISE) curve. This curve plots the False Negative Rate (good ideas wrongly sacrificed) against the percentage of all ideas screened out (see Figure 1, b and d). The maximal distance between both is the optimal screening rate.

To identify an AI model for idea screening, this paper tests the out-of-sample performance of eight model specifications for idea selection, including those from three previously published theory-based models: Word Colocation (Toubia and Netzer 2017), Topic Atypicality (Berger and Packard 2018), and Inspiration Redundancy (Stephen, Zubcsek, and Goldenberg (2016). Here we provide a brief intuition for each theory-based model but detail them in the Theory section. The intuition of Word Colocation is that good ideas balance novelty and familiarity. The intuition of Topic Atypicality (designed for song lyrics) is that good ideas differ from other ideas (the typical) in the same contest. The intuition of Inspiration Redundancy is that good ideas come from ideators with diverse connections, who provide less redundant sources of inspiration.

We test the models with their original predictors and new ones we develop. Most importantly, we test these models, *out-of-sample,* where in their original exposition, they were tested in-sample. In-sample testing may exploit sample idiosyncrasies rather than underlying patterns of creativity. We use four methods for testing the models: the **l**east **a**verage **s**hrinkage and **s**election **o**perator (LASSO), Bayesian Stacking, Random Forest, and RuleFit. We also tested multi-arm bandits (see Web Appendices 1 and 2).

The specific goals of this paper are the following. 1) Asses how AI can assist managers in idea screening by studying to what extent one can replace expert evaluations with models. In particular, we compare the performance of three published theory-based models using out-of-sample prediction. 2) Identify simple models and predictors for idea screening, if any. 3) Compare out-of-sample prediction performance of four methods: LASSO, Bayesian Stacking, Random Forest, and RuleFit for testing models. We test the models' performance on 21 different real-world crowdsourcing contests conducted for large firms. The pooled data contains 4191 ideas from 1467 ideators. We also test on a 22nd held out contest with internal and external experts.

This study has three major findings. First, while even the best previously published theory-based models cannot mimic human experts in choosing the best ideas, a simple model using LASSO can efficiently screen out ideas considered bad by experts. In an additional 22nd hold-out contest with internal and external experts, the simple model does better than external experts in predicting the ideas selected by internal experts. Second, the authors develop an Idea Screening Efficiency curve that trades off the False Negative Rate against the total ideas screened. Managers can choose the desired point on this curve given their loss function. The best model specification can screen out 44% of ideas, while sacrificing only 14% of good ideas. Alternatively, for those unwilling to lose any winners, a novel two-step approach screens out 21% of ideas without sacrificing a single 1st-place-winner. Third, a new predictor, Word Atypicality, is simple and efficient in screening. Theoretically, this predictor screens out atypical ideas and keeps inclusive and rich ideas. These three findings provide methodological, substantive, and managerial contributions respectively to the literature on ideation. The rest of the paper is in the following seven sections: literature, model, data, method, results, analysis of extended dataset, and discussion.

2. Literature

Our work relates to the literature on crowdsourcing (Allen, Chandrasekaran, and Basuroy 2018; Stephen, Zubcsek, and Goldenberg 2016; Toubia and Netzer 2017), where screening large numbers of ideas can provide huge efficiencies. The three theoretical models of interest are: Word Colocation (Toubia and Netzer 2017), Topic Atypicality (Berger and Packard 2018), and Inspiration Redundancy (Stephen, Zubcsek, and Goldenberg 2016). Table 1 provides an overview of these original models, our extensions of them, and their respective intuitions. We first review the theory underlying these models to guide our research.

2.1 Word Colocation

The internet contains a large amount of freely accessible text. One important contribution of Toubia and Netzer (2017) is to show how publicly available data can be used to potentially automate idea evaluation. Their metrics access "global" information (i.e., information not specific to the evaluation context) to assess idea quality. To apply this information, e.g., to evaluate ideas, individuals need to categorize this information. Prototype theory (e.g., Mervis and Rosch 1981), which draws on the concept of atypicality in semantic categories (Rosch, Simpson, and Miller 1976), provides a good categorization approach.

Novelty Versus Familiarity. Many new ideas and concepts are the outcome of a process of combination and reorganization of existing ideas and concepts (Mobley et al. 1992). Innovativeness consists of reassembling elements from existing knowledge bases in a novel fashion (Dahl and Moreau 2002). Thereby, within an idea, a moderate level of incongruity between the concepts that make up the idea can be beneficial (Finke et al. 1992). Research has identified a theory about optimal levels of incongruity: the concept of familiarity versus novelty (Toubia and Netzer 2017) draws on the cognitive perspective of innovativeness (Dahl, Chattopadhyay, and Gorn 1999; Goldenberg and Mazursky 2002), which asserts that evaluators rely on information stored in their memories to judge ideas (Toubia and Netzer 2017). When individuals perceive stimuli related to their knowledge, the stimuli activate their domain-specific schemas (Bilalić et al. 2008). Because of schema activation, Toubia and Netzer (2017) argue that if an idea is too novel, its evaluation takes place largely in a vacuum and the evaluator will not

know how to judge it; if an idea is too familiar, it seems to be rather incremental – or not new or interesting at all. Thus, experts rank highest those ideas that optimally balance novelty and familiarity based on their current knowledge.

With modern text-mining methods, word colocation networks can be constructed in seconds. In a word colocation network, the vertices are words (or technically word stems or word lemmas), and edges indicate co-occurrence. Words that appear together more frequently have higher edge weights and are therefore "closer" to each other (Netzer et al. 2012). Toubia and Netzer (2017) use the group of edge weights in an idea to measure its balance of "novelty" compared to "familiarity." A key point is that it is not the words that directly determine novelty or familiarity, but instead, the combinations of words within an idea. Thus, for Toubia and Netzer, an idea is novel to the degree it contains words that typically do not appear together. It is familiar to the degree that it contains words that frequently appear together.

Toubia and Netzer (2017) type the problem description of the contest in Google and use Google Search results to construct a word colocation network. This approach implicitly assumes that Google Search results represent the popularity of the respective website and its content among the crowds because the search results are rank ordered based on the activity of a very large group of users. Thus, a word colocation network computed from averaging over high ranking (roughly top 50) results, provides information on whether ideas that use certain pairwise word combinations are thought to contain a desirable balance between "novelty" and "familiarity."

2.2 Topic Atypicality

Atypicality is a construct that deals with the uniqueness of an idea relative to a set of ideas (Berger and Packard 2018). Besides innovativeness, the communication of an idea is crucial for success (Kilgour, Koslow, and O'Connor 2020; Runco 1995). Poor communication of an idea makes it hard for external experts to see the idea's merit (Simonton 1999).

The question is whether atypicality is positively or negatively related to idea quality. The literature provides evidence for both. On the one hand, some research suggests that in music, a creative context, songs with atypical lyrics, i.e., lyrics which diverge in content from a genre average, are more

likely to become successful (Berger and Packard 2018), because novelty or atypicality can increase attention, evaluation, and liking (Berlyne 1970; Berger and Packard 2018)[1]. If we apply this logic to our setting of idea screening, experts may prefer ideas that are atypical or differentiated from others. On the other hand, other research suggests that genre-typical creative content tends to have a higher quality (Lamb, Brown, and Clarke 2015; Ritchie 2001). Evaluations are better if the communication is clear and complete (Dean et al. 2006), includes a lot of details (Durand and vanHuss 1992), or elaborated, i.e., understandable, complete, and contains many elements (Besemer and Treffinger 1981). The link is that completeness assists comprehension, which leads to higher judgments of the idea (Sukhov 2018). Research in ideation finds that if ideators independently come up with similar ideas to a given problem, these typical ideas tend to be better. The reason is that these less atypical, i.e., more common ideas, may indicate a widely held need, which indicates market acceptance of the innovation; this leads to the fact that more typical ideas tend to have a higher value (Kornish and Ulrich 2011).

2.3 Inspiration Redundancy

The key idea of interconnectivity is to consider the influence on the ideator through a network. If ideas submitted to crowdsourcing contests are visible to other participants, it can inspire and potentially influence them (Wooten and Ulrich 2019). The network structure that surrounds ideators can provide information about the redundancy of their inspirations – which in turn influences the quality of the ideas they submit (Stephen, Zubcsek, and Goldenberg 2016).

If the ideator's network neighbors are not connected to one another, ideators receive independent inspirations. In contrast, if the network neighbors are connected, they also influence each other and ideators receive similar, redundant inspirations (Burt 2004). Stephen, Zubcsek, and Goldenberg (2016) name the following reasons why higher redundancy leads to lower quality ideas: (1) a decreasing size of the set of neighbors' ideas which serve as inspirations when ideating may lead to decreasing

[1] We do not use Berger and Packard's Latent Dirichlet Analysis (LDA) based approach since it extracts the topics (dimensions), e.g., bundles of words, that are most popular (common). The LDA approach can be problematic in the context of new product ideas because LDA may categorize novel and unique words as "errors." Successful new product ideas tend to be novel or unique (Dahl and Moreau 2002; Toubia 2006). In crowdsourcing contests for ideation, a metric that captures atypicality at the idea-level instead of the topic-level, may be superior because it does not screen out such novel or unique ideas.

innovativeness, (2) idea redundancy could stifle individual innovativeness because it interferes with psychological mechanisms like fixation (Bayus 2013) involved in processing others' ideas, and (3) the recurrence of an idea operates as a proof signal. These mechanisms may lead to similar ideas and to a decrease in variance of idea quality in the contest. However, a high variance of idea quality is desirable because it increases the likelihood of finding a few outstanding ideas (Terwiesch and Ulrich 2009). Thus, high interconnectivity may relate negatively to idea quality.

3. Theoretical Models

This section describes how we operationalize the three theoretical models: Word Colocation, Topic Atypicality, and Inspiration Redundancy.[2] Two of the theoretical models apply to the text of ideas and the third one is based on the ideator's commenting network. Each theoretical model provides a key metric: the Kolmogorov-Smirnov distance (to measure Word Colocation; Toubia and Netzer 2017); Peer Deviation LDA (to measure Topic Atypicality; Berger and Packard 2018); and Clustering Coefficient (to measure Inspiration Redundancy; Stephen, Zubcsek, and Goldenberg 2016).

3.1 Text Mining

Text mining (Berger et al. 2020; Netzer et al. 2012; Netzer, Lemaire and Herzenstein 2019) has become increasingly popular as a tool since it helps to detect patterns in large unstructured text corpora, by which one can generate knowledge about consumers (Matz and Netzer 2017; Wedel and Kannan 2016).

Text Pre-Processing. In each case, we prepare the text by eliminating "stopwords," or words which appear extremely commonly (e.g., "the" and "and"). We run a process called lemmatization which stands in place of the traditional approach of stemming. While stemming simply truncates words to reduce duplicates of the same word, lemmatization attempts to remove inflectional endings and reduce words to a base dictionary word. Word stems are not always words, while word lemmas always are. We chose lemmatization over stemming mainly for convenience in working with the text since it is easier to

[2] We attempt to operationalize the theoretical models in the same way as the published research when possible, but in some cases our implementation differs slightly because of context.

make sense of word lemmas than word stems. The impact on downstream performance is likely to be very small but lemmatization may be slightly better (Balakrisnan and Lloyd-Yemoh 2014).

Word Colocation. We apply the work of Toubia and Netzer (2017), which uses several metrics computed from the words used to describe an idea. The main building blocks for these various metrics are word frequencies and Jaccard indices. Both, word frequencies and Jaccard indices require a reference corpus and a word colocation network to be computed. As mentioned above, Toubia and Netzer (2017) introduced a novel reference corpus consisting of the first 50 Google results when the ideation topic is entered as a search term. The word frequencies are simply the number of times each word in the idea appears in the reference corpus. The Jaccard index, computed on a word *pair*, is the intersection over the union of documents containing the respective words in the pair. Here, the word "document" refers generally to a body of text. In practice, the researcher specifies the documents. If j and k are words and D_j and D_k are sets of documents containing them, respectively, then the Jaccard index between *j* and *k* is:

$$J(j,k) = \frac{D_j \cap D_k}{D_j \cup D_k} \quad 1$$

For example, if the documents are "one two three," "one two," and "one," then the Jaccard index between the words "one" and "three" is 1/3, since one document contains both while all three contain at least one of the two. With the Jaccard indices and the word frequencies from each idea, Toubia and Netzer (2017) construct several metrics: the average, max, and min word frequency of an idea; the average max, and min Jaccard indices from an idea; the coefficient of variation of word frequencies; and the coefficient of variation of Jaccard indices. A key metric is the Kolmogorov-Smirnov (KS) distance, which is created by first computing Empirical Cumulative Distribution Functions (ECDFs) for each document of a reference corpus, taking the average of those results, and then comparing the ECDF of a given idea to the average using the KS distance. The KS distance is the maximum absolute difference between two vectors. Toubia and Netzer show that the KS distance, their metric to balance novelty and familiarity, relates negatively to idea quality even after controlling for other word-derived metrics.

Because Jaccard indices measure how often word pairs are collocated, we view the KS distance as capturing the idea's Word Colocation.

Using each reference corpus, we compute the metrics used in Toubia and Netzer (2017): the mean, min, max, and coefficient of variation for both Jaccard indices and word frequencies of each idea, and the KS distance for each idea.

Topic Atypicality. This model is developed in the spirit of Berger and Packard's (2018) model of Content Atypicality. Their application is further removed from our ideation context compared to the other two theory-based models: they study innovativeness in a music setting. The dependent variable is song popularity. Berger and Packard use Latent Dirichlet Allocation (LDA) to generate a topic model from a corpus of song lyrics. LDA "is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities." (Blei et al. 2003). In other words, LDA views topics as having a relationship with individual words: some words are likely or unlikely to appear for certain topics. A common representation for LDA model outputs is a set of linear equations, one for each topic, where the equation terms represent words. Each word term has a coefficient indicating the importance of that word to the topic. These equations can be inputs into operations. Specifically, a topic score can be computed for a document by applying the model weights against each word in the document. Thinking of the topics as coordinates, each document can be imagined as having a 'location' somewhere in topic space. In our case, we create an LDA model for the entire set of ideas in a contest and compute its topic location by applying the LDA weights to its vocabulary. Then for each idea we compute the distance between it and the location of the overall corpus in topic space.

Berger and Packard use Ireland and Pennebaker's (2010) style matching formula to measure distance, which is sensitive to the degree of deviation within each topic. Define the "topic function," l, as a mapping from a collection of words into a particular topic. For LDA models, each topic function is simply a weighted average of indicator variables for specific words. Index ideas are represented by i, with reference corpus C. Then for a single topic, l, the Topic Atypicality $A_{TA}(C, l, i)$ is given by

$$A_{TA}(C, l, i) = \frac{1}{1 - |l(C) - l(i)|/(l(C) + l(i) + .001)} \quad 2$$

The total Topic Atypicality, $A_{TA}(C, i)$ is calculated by taking the average of each $A_{TA}(C, l, i)$ over topics *l*. Mathematically, if N_l is the number of topics:

$$A_{TA}(C, i) = \frac{\sum_l A_{TA}(C, l, i)}{N_l} \quad 3$$

In addition to Topic Atypicality, we develop a new metric, Word Atypicality A_{WA}. This metric counts the number of words in common between an idea and a reference corpus *C* divided by the total vocabulary size of *C* and deducts this value from 1. The intuition of Word Atypicality is that it identifies ideas with unique words; ideas with many unique words have higher values of Word Atypicality. Ideas which do not contain unique words will have a score of 0. Mathematically, if W_C is the set of words in *C,* and W_i the set of words in *i*, then Word Atypicality is:

$$A_{WA}(C, i) = 1 - \frac{W_C \cap W_i}{W_C \cup W_i} \quad 4$$

which is simply one minus the Jaccard index between the word sets. In realistic settings, W_c will be a much larger set than W_i, which tends to make the second term small. This means that Word Atypicality, in turn, is often near 1.

Two relevant differences distinguish Word Atypicality from Topic Atypicality. First, Word Atypicality is at the level of words, while Topic Atypicality is specified at the topic level. Topics in LDA come from general patterns in the corpus and so are less sensitive or insensitive to unique words. Unique words may well fall into less meaningful topics or be screened out all together from the analysis. Word Atypicality, on the other hand, is exclusively sensitive to unique words. The second difference is that Word Atypicality measures deviations as binary, either a word overlaps or does not. Topic Atypicality measures the degree of deviations as a continuous distance. Topic Atypicality is sensitive to the number of times a particular word is used while Word Atypicality is not. This feature may mean that Topic

Atypicality is susceptible to noisy outliers which use certain words with high LDA-topic weights, where Word Atypicality would not be. As Word Atypicality ends up featuring centrally in the important results, Web Appendix 3 provides some examples that facilitates its understanding.

3.2 Network Metrics

Recent work has used social network data to study innovativeness in ideation (Stephen, Zubcsek, and Goldenberg 2016; Godart and Galunic, 2019). We explain the two metrics developed using networks.

Inspiration Redundancy. This model is inspired by Stephen, Zubcsek, and Goldenberg (2016). For each ideator, we create a network where nodes are ideators and links are comments. If one ideator has made a comment on another ideator's idea, those two ideators are connected. Otherwise, they are not connected. Using this network, we compute the Clustering Coefficient (Watts and Strogatz 1998) for each ideator. The Clustering Coefficient for a node *n* in an undirected graph is the number of edges among n's neighbors that exist divided by the number that could exist. If N_n is the "neighborhood" of *n, m* and *p* are neighbors of *n,* e_{mp} is an edge between *m* and *p*, v_m and v_p are nodes for *m* and *p,* and node *n* has k_n total neighbors, the original Clustering Coefficient CC_n is:

$$CC(n, N_n) = \frac{2\left|\{e_{mp}: v_m, v_p \in N_n\}\right|}{k_n(k_n - 1)} \quad 5$$

Since the unit of analysis is ideas, we then apply the Clustering Coefficient of each ideator to all his or her ideas. This Clustering Coefficient is computed for the group of comments at the very end of each contest. We view the Clustering Coefficient as measuring the ideator's Inspiration Redundancy because it captures the degree of connectivity of each ideator's sub-community. Since comments reflect the information present in the network surrounding an ideator, the Clustering Coefficient measures the diversity of information around the ideator. Stephen, Zubcsek, and Goldenberg (2016) showed in their context that higher Clustering Coefficients related negatively to idea innovativeness as judged by consumers. The reason is that high clustering indicates that the inspiration sources in that area of the network are less diverse. This original version of the Clustering Coefficient is calculated at the end of the

contest but *prior to idea evaluation by the experts*. Thereby, it makes full use of all information available at the point of evaluating the ideas at the end of the contest. From a managerial perspective, it is feasible to implement (in fact it is the simplest approach), and it has the potential to predict well.

This original approach to calculating the Clustering Coefficient may be subject to endogeneity, because it considers information that is available after idea submission.[3] To address this limitation, we implement a modified version of the Clustering Coefficient, which considers information available only until the time of submission, and only uses outdegree. If $N_n(t)$ is the "neighborhood" of *n* at time t, *m* and *p* are neighbors of *n*, e_{mp} is an (outgoing) edge between *m* and *p*, v_m and v_p are nodes for *m* and *p*, and node *n* has $k_n(t)$ total neighbors at time t, the modified Clustering Coefficient MCC_n at time t is:

$$MCC(n, N_n(t), t) = \frac{2\left|\{e_{mp}: v_m, v_p \in N_n(t)\}\right|}{k_n(t)(k_n(t)-1)} \quad 6$$

We also use the Constraints metric (Burt 2009). This metric has been used to measure the popularity of cultural elements in fashion (Godart and Galunic 2019). The Clustering Coefficient only considers information from the direct neighbors, but the Constraints metric also considers second order neighbors (i.e., neighbors of neighbors), which is particularly helpful in the context of sparse networks early in the contests. The Constraints metric reflects the value of a particular node as a "bridge" between nodes which are otherwise rarely bridged. In the context of innovativeness, connections between concepts, people, or facts which are not typically connected have higher potential for novelty.

Let a_{nm} indicate the existence of an outgoing link from *n* to *m,* so it equals l if ideator *n* commented on *m's idea*, and 0 otherwise. Then define $\rho_{nm}(t) = a_{nm}/N_n(t)$, where $N_n(t)$ is the neighborhood of n at time t, just as above. Then the Constraints metric is

$$C(i, N_n(t), t) = \sum_{m \neq n} (\rho_{nm}(t) + \sum_{p \neq n; p \neq m} \rho_{np}(t)\rho_{pm}(t))^2 \quad 7$$

[3] This concern is not relevant in the experimental approach by Stephen, Zubcsek, and Goldenberg (2016), in which the ideators' network position was induced by the experimental design, but it may become problematic in our study which uses field data when the comment network evolves over time. Additionally, we use comments as a proxy for exposure, while the Stephen, Zubcsek, and Goldenberg (2016) paper uses real links between individuals.

This metric sums across neighbors of a node *n*. For each (outgoing) neighbor *m*, the Constraints metric for node *n* is larger if *n* and *m* have many neighbors in common. The metric is also larger if *n* has fewer connections. We expect this metric to be negatively related to idea quality, as larger values indicate that node *n* has fewer connections and connects fewer groups of otherwise unconnected nodes.

4. Data

4.1 Data Source and Context

Our data consists of 21 different crowdsourcing contests that a crowdsourcing platform, Hyve, conducted for large corporate clients. Web Appendix 4 provides an overview of the contests. The client firms were Frankfurt Airport, Lufthansa, MasterCard, Deloitte, Telekom, Vodafone, Zeiss, Volkswagen, and DHL. Typically, both, Hyve and its clients recruit ideators by public announcements and privately contacting past ideators. The contest usually answers one specific question, runs for a limited time, between 30 and 80 days, with an announced deadline. All contests are idea generation contests that search for innovative ideas about future products, services, or business models. We only use data from contests in which the ideas are verbally described. This excludes another popular form of contest, design contests (cf. Allen, Chandrasekaran and Basuroy 2018). The contests run on the same platform and offer social networking functions: ideators can explore, evaluate, and comment on the ideas of others. The platform does not allow formal collaboration. The idea evaluation occurs in three stages: experts rate all ideas, experts select a shortlist of ideas to be presented to the jury of client's managers, jury selects winners.

Expert Ratings. After the contest is over, experts on the topic, usually from the client company, evaluate all ideas on a 5-point scale.

Experts' Shortlist of Ideas. Next, based on their own evaluations, the experts build up a shortlist of 12 to 30 ideas. Usually, the experts' top-rated ideas make the shortlist. However, if one expert likes a specific idea well, he or she can discuss it with other experts; if those do not oppose, such an idea can additionally make the shortlist. All shortlisted ideas usually receive at least a small prize or some formal recognition.

Jury's Selection of Winners. Finally, the shortlist of ideas as well as a contest overview is presented to a jury of 5 to 10 members which usually consists of client's top executives, experienced innovators, professors, or consultants on the topic of the contest. The jury selects winners in one of two ways. Either the jury members individually vote on the ideas using a scorecard tailored to specific innovation criteria, a simple aggregation determines the winners; or the jury jointly selects the winners in a discussion. In addition to the ideas proposed by the experts, jury members have the option to pick and evaluate any idea from the contest in a discussion session.

4.2 Dependent Variable and Predictors

This section provides a description of the dependent variable and independent variables.

Dependent Variable: Success. The dependent variable is success. We consider an idea a success if it makes the experts' shortlist. We choose this shortlist as the success metric for three reasons:

First, every shortlisted idea receives some prize or formal recognition, which means, a reward. We consider receiving rewards to be an indication of success.

Second, in private discussions, managers of Hyve stated that they have a high degree of confidence in the shortlist, but not in the winner.

Third, the juries select very few winners relative to the thousands of ideas. Ten contests had a total of 12 winners, some contests had more than one winner. Such a rare event in the dependent variable results in a low variance – which models typically cannot capture in a meaningful way. In contrast, the experts' shortlists consist of up to 30 ideas and, thus, are a richer dependent variable than the jury's winners. So, we code shortlisted ideas as 1, other ideas as 0.

Predictors. Table 2 lists the independent variables, which we explain below.

The three theoretical models have sets of metrics which we include as predictors in our model specifications. Word Colocation yields these predictors: max, min, mean, coefficient of variation for Jaccard indices and node frequencies, along with the Kolmogorov-Smirnov distance. Based on Topic Atypicality, we develop a predictor, Word Atypicality. The reference corpus for both is the set of other ideas in the same contest as the focal idea. We also create a variant of Word Atypicality where the

reference corpus is the Google Search results. The Inspiration Redundancy model yields these predictors: ideator degree, Clustering Coefficient, and Burt's Constraints metric.

Our goal is to screen ideas out-of-sample, where out-of-sample refers estimating on 20 contest and predicting on the 21st. Since each out-of-sample prediction task involves a single contest, any contest-level variables will be constant across ideas. In other words, contest-level predictors cannot distinguish between ideas within a contest. Therefore, we do not include contest-level variables. Since we need to control for differing numbers of shortlisted ideas and total ideas across contests, we include the ratio of shortlisted ideas to total ideas in each contest as control. This variable is not absorbed in the global intercept and it is usable for out-of-sample prediction.

Despite spending months comparing numerous models and methods, we only present the coefficients of two models in Section 6. First, for consistency with prior literature, we present the results of a model that only contains the three original predictors based on literature. Second, we present the results of a model that contains all predictors from Table 2.

5. Method

Inspired by the above theoretical models, we test 8 model specifications, using 14 predictors and four methods, on 4191 ideas from 21 contests. The 8 model specifications include predictors from each of the above theoretical models, both alone and separately, along with some new predictors developed here.

The three models discussed above each have one central predictor. Word Colocation has the Kolmogorov-Smirnov distance (Toubia and Netzer 2017), Inspiration Redundancy has the Clustering Coefficient (Stephen, Zubcsek and Goldenberg 2016), and Topic Atypicality has the Latent Dirichlet Allocation topic distance from (Berger and Packard 2018). We develop a new predictor called Word Atypicality, inspired by Topic atypicality. We use Word Count (number of words in an idea, including stop words) as a naïve benchmark. We use additional predictors which were either originally included as controls in the papers publishing the three theoretical models, or extensions of those theoretical models. Fourteen predictors appear in at least one model specification. Table 2 shows all fourteen predictors.

The 8 model specifications are composed of various combinations of predictors. Three specifications have one predictor, which is the central predictor from the three theoretical models. A separate model specification uses all three. Word Atypicality with two different reference corpora (ideas from the same contest and Google Search) and Word Count add three more model specifications. The final model specification includes all fourteen predictors.

Overall, we train (fit) each model specification on 20 contests and determine performance on the one contest held out. So, we have 21 iterations for each model specification and method. This testing amounts to 588 runs (21 contests x 7 specifications x 4 methods). For predictive rigor, all our testing is out-of-sample and cross-validated.

5.1 Model Specification, Fitting, and Prediction

For all model specifications, the dependent variable is binary, whether an idea is on the shortlist or not:

$$y_i = \begin{cases} 1 \text{ if idea } i \text{ is shortlisted} \\ 0 \text{ otherwise.} \end{cases} \quad (8)$$

The models vary in specification, depending on which set of the 14 predictors from Table 2 are used. The predictors are characteristics of ideas, ideators, and contests. We do in-sample fitting to estimate the relative standardized coefficients and out-of-sample fitting to ascertain relative performance of models.

To find the most parsimonious set of predictors, we use the Least Average Shrinkage And Selection Operator (LASSO), a statistical and AI method (Tibshirani 1996; see also Rafieian and Yoganarasimhan 2021). LASSO has robust performance across many entirely different settings (Abadie and Kasy 2019). This feature makes LASSO appealing for our context because we need predictor variables that are robust across contests on varying topics for entirely different clients, with distinct judging panels. Web Appendix 5 presents details on LASSO.

Here we briefly summarize the procedure used to fit the models. We use an outer cross-validation loop and an inner cross-validation loop. The outer loop consists of three steps: designate one contest as the holdout, train the model on the remaining 20 contests, make predictions for the holdout. The inner loop uses cross-validation to set the LASSO penalty parameter which controls parsimony. This inner loop

cross-validation operates on 20 contests. The inner loop and outer loop connect through the tuning parameter: for each holdout contest in the outer loop, a corresponding inner loop is used to find the best setting of the tuning parameter (on the non-holdout contests) and make predictions for the holdout. Web Appendix 6 describes this cross-validation procedure step-by-step.

5.2 ROC Curve and Idea Screening Efficiency Curve

ROC Curve and AUC. Removing "bad" ideas requires a high degree of sensitivity (in the technical sense: $\frac{TP}{TP+FN}$; TP = true positives, FN = false negatives) in order not to accidently remove "good" ideas. Main criterion for predictive accuracy, out-of-sample, is the Receiver Operating Characteristic or **ROC** curve. The ROC curve is commonly used in the Computer Science and Information Systems literatures. It plots the false positive rate (x-axis) versus the true positive rate (y-axis) for values between 0 and 1. (See Figure 1, a and c). Our main criterion of predictive accuracy out-of-sample is the **a**rea **u**nder the ROC **c**urve (AUC). It provides information on the goodness of fit of the model, whereby .5 indicates not better than random, while higher values indicate increasing levels of the model's goodness of fit. AUCs between .7 and .8 indicate acceptable fit, AUCs above .8 excellent fit (Hosmer and Lemeshow 2013, p. 177).

Idea Screening Efficiency Curve. In idea screening, Hyve´s clients previously had to choose ex-ante whether they wanted to minimize false negatives (retain all good ideas), which comes at the cost of high screening effort (potentially very high), or whether they wanted to screen out false positives (bad ideas), which may come at the sacrifice of eliminating some good ideas. Thereby, the exact preference differed between clients. As an additional flexible criterion, we develop the Idea Screening Efficiency curve, which we plot against the threshold of acceptable performance provided by Hyve's director. This was done to screen out 25% (50%) of bad ideas without sacrificing more than 15% (30%) of good ideas. The Idea Screening Efficiency (ISE) curve is a plot of the percentage of all ideas screened out, on the x-axis, against the False Negative Rate (FNR) on the y-axis (see Figure 1, b and d). The ISE curve has the same shape as the ROC curve (though rotated 180˚) but is presented with axes which are more directly interpretable within the context of our problem. The false negative rate is:

$$1 - sensitivity = \frac{FN}{FN + TP} \quad (9)$$

In words, it is the percent of shortlisted ideas that are falsely predicted to be non-shortlisted ideas.

The ISE curve is a flexible tool that provides information about all possible tradeoffs between any given reduction in screening effort (ideas) versus the respective sacrifice of good ideas. This stands in contrast to methods such as rare events logistic regression (King and Zeng 2001) which applies alternate cutoffs for binary classification, when the distribution of positives and negatives is unbalanced. Because the ISE curve plots results of all possible cutoffs, any decision make can choose the cutoff that optimizes his or her tradeoff between false negatives and all ideas screened. Beyond crowdsourcing, this ISE curve is useful for any predictive exercise with high imbalance between positives and negatives, where decision makers differ in their loss functions. It provides a simple elegant visual to trade off false negatives and false positives. Such prospects could be ideas, potential consumers, potential new products, or proposals.

5.3 Reference Methods

We also test LASSO against three other methods proposed by or inspired by the reviewers: Random Forest, Rule Fit, and Bayesian Stacking. Like Lasso, Random Forest (Breimann 2001) generally performs well on tabular data and resists overfitting. RuleFit (Friedman and Popescu, 2008) is a combination of Random Forest and LASSO. Bayesian Stacking (Yao et al. 2018) constructs a weighted average across different models. Details of each of these methods are in Appendices 7, 8, and 9.

6. Results

Our summary results are the following. First, current models are unable to replace humans in selecting the best ideas. Second, however, these models do an excellent job in screening bad ideas, reducing experts' tedium and enabling their focus on the best ideas. Third, the authors develop an Idea Screening Efficiency curve that relates the False Negative Rate of good ideas screened out with the rate of ideas screened. Managers can choose the desired point on this curve for optimal idea screening. For example, the best model specification can screen out as much as 44% of bad ideas sacrificing only 14% of good ideas. A

two-step model screens out 21% of the worst ideas without sacrificing a single 1st-place-winner. Fourth, a new predictor, Word Atypicality, is simple and efficient in such screening. Theoretically, this predictor screens out atypical ideas and keeps inclusive and rich ideas, which experts rated high. Word count is simpler but does not perform as well.

Detailed results follow.

6.1 AUC-Curve: Results

In-sample Estimates of Coefficients. To appreciate effect sizes of coefficients, we first test in-sample the specification that includes only three theory-based predictors, pooling all 21 contests. (Out-of-sample tests would yield 21 sets of coefficients, one for each contest held out.) To control for contest heterogeneity when pooling contests, we also include the percentage of shortlisted ideas of each contest (% Shortlisted). This control is different for each contest but same for each idea within a contest.

Table 3 shows the standardized coefficients of the LASSO logistic regression estimated in-sample for the pooled 21 contests. The model specification includes the three theory-based predictors. LASSO retains all three. *This result means the three are complimentary, each capturing one unique dimension of the innovativeness of an idea*. The largest standardized coefficient is on the Clustering Coefficient with a value of -.19. The next largest is on the Kolmogorov-Smirnov distance calculated on the Google reference network with a value of -.08. The third is on Topic Atypicality with a value of -.07. Notably, the signs of the first two coefficients are in the direction predicted by their original theory. However, the sign for the third, Topic Atypicality, is opposite to that in the original application (music). The reason may be due to different dependent variables (attention getting versus creative) and contexts (music versus ideation). In music, novelty is most attention getting and so the most atypical song is rated highest (sign is positive). In ideation, the most detailed and comprehensive idea (most inclusive or "typical") is rated highest.

Out-of-sample Predictive Performance. For predictive rigor, we test the same model specification with the three theory-based predictors out-of-sample with cross-validation. Web Appendix 6 explains our method for out-of-sample predictions. Figure 1a shows that the AUC of the ROC curve has a value of .72.

Thus, the three original predictors from the theory-based models jointly reach a threshold which is generally considered to be acceptable.

Next, we test the model specification with all 14 predictors from Table 2. Importantly, LASSO retains only Word Atypicality as a predictor in all 21 contests, sometimes complemented by another predictor. Web Appendix 10 contains some additional information about the performance of Word Atypicality. Figure 1c shows the out-of-sample ROC curve. The AUC is .73, which is a little over the value above of .72, even though it (usually) contains only 1 predictor. This result has two important implications. One, *that the new predictor we developed, Word Atypicality, is more powerful in prediction than any other predictor, singly or in combination. Two, Word Atypicality encompasses the predictor capacity of all three theory-based predictors.*

6.2 LASSO's Comparison to Other Methods

We next compare the above out-of-sample results with LASSO to results using three other methods, RuleFit, Random Forest, and Bayesian Stacking, for the model specification which includes all 14 predictors. Web Appendices 7 to 9 provide some additional information on modeling and key results. Random Forest has an AUC of .69, RuleFit of .70, and Bayesian Stacking of .72. Overall, LASSO does better than the other three methods. Also, the optima of the Idea Screening Efficiency curves for Random Forest, RuleFit, and Bayesian stacking exceed the performance threshold from Hyve by smaller amounts.

Recall that Bayesian Stacking is an ensemble which creates a weighted average of models. Bayesian Stacking finds the weights of each model by using **l**eave **o**ne **o**ut (LOO) cross validation. In the case of Bayesian Stacking, we tried creating ensembles from many different models formed by selecting random subsets of predictors. The best configuration we find is an ensemble over three model specifications. Model Specification 1 uses the KS Distance from Word Colocation and the Clustering Coefficient from Inspiration Redundancy as predictors, Model Specification 2 uses Word Count alone, and Model Specification 3 uses Topic Atypicality and Word Atypicality. The combination of the predictions from these three model specifications via Bayesian Stacking reaches an AUC of .72. These results compare to LASSO's AUC of .73.

6.3 Optimal Screening Rate on the Idea Screening Efficiency Curve

Figure 1b shows the ISE curve in blue. Figure 1b shows it for the model specification with the three theory-based predictors. Figure 1d shows it for the model specification with all predictors from Table 2. The green dotted line shows the managerial threshold given by Hyve: screening 25% of all ideas without losing more than 15% of good ideas or screening 50% of ideas without losing more than 30% of good ideas. In Figures 1b and d, Hyve's standard is met anywhere the solid blue curve falls below the dotted green line and the optimum point is when the blue curve is maximally below the green dotted line.

Table 4 shows the optimal screening rate for all eight model specifications. For our data and the model specification with all predictors, the best performance screens out 44% of all ideas at the cost of sacrificing only 14% of good ideas. This result *exceeds* Hyve´s standard and indicates that our model can provide a substantial reduction in experts' workload.

The code to run this model is in Web Appendix 13.

6.4 Theory-Based Predictors and Word Atypicality: Substitutes or Complements?

Preliminary results based in Section 6.1 indicate that the three theory-based predictors tend to be complementary, but that Word Atypicality tends to encompass all three. We analyze the intersection of sets of ideas predicted by various predictors at the top and the bottom to further explore complementarity. For this purpose, we use a method, new to marketing, called the Super Exact Test for Efficient Testing of Multi-Set Interactions (Wang, Zhao, and Zhang 2015). This method allows for testing the statistical significance of the size of overlap between multiple sets. For each predictor, i.e., the three theory-based predictors and Word Atypicality, we divide the predicted ideas into three categories: top 25% of predicted ideas (to retain), bottom 25% of predicted ideas (to screen out), remaining predicted ideas. Then, we compare the intersections between the top and bottom 25% of predicted ideas by all four predictors.

Table 5 contains the results for overlap at the top (select) and bottom (screen). By chance, the pairwise overlap in sets classified by any two predictors would be 25% x 25% = 6.25%. The pairwise overlap between the three theory-based predictors, Word Colocation, Inspiration Redundancy, and Topic Atypicality, indicates that no pair significantly overlaps at the top and at the bottom. This additional result

confirms that the three theory-based predictors are unique, each capturing different dimensions of what experts consider to be good ideas. So, they are complementary. On the other hand, the ideas classified by Word Atypicality overlap significantly with *each* of the three original models at the top and at the bottom. This result confirms that even though Word Atypicality is parsimonious, it partly captures dimensions that are unique to the three theory-based predictors. Thus, it is a substitute to the theory-based predictors.

7. Managerial Relevance

The key to applicability of a model is its relevance for managers. To address this issue, we carry out four additional analyses: screening ideas without losing winners (Section 7.1), a new 22nd contest with multiple internal and external ratings (Section 7.2), and the relationship between theory-based predictors and managerial ratings (Section 7.3). Details of each of these analyses follow.

7.1 Two-Step Approach

Two-Step Approach. The goal of the two-step approach is to screen out the worst ideas without losing a winner. Step 1 scores each idea within each contest on a simple heuristic and screens out ideas that score the lowest. Step 2 runs the best predictive model from Table 4 on the reduced corpus of ideas.

Summary of Step 1. (Details in Web Appendix 11) We rank-order all ideas on one or more predictors. We test one predictor at a time, or combinations of two or three predictors from the 14 predictors in Table 2. We then screen out ideas that fall below a threshold to reduce noise in the corpus of ideas. In the case of one predictor, we screen out the worst ideas on that predictor. In the case of two predictors, we screen out the worst ideas on the rankings of *both* predictors (i.e., the bottom intersection of two rankings of ideas). In the case of three predictors, we screen out the worst ideas on the rankings of *all three* predictors (i.e., the bottom intersection of three rankings of ideas). Testing one predictor at a time requires 14 runs. Testing 2 predictors at a time requires 91 runs. Testing 3 predictors at a time requires 364 runs. In total, this ranking exercise requires 469 runs. We also test various thresholds for screening out from 5% to 35% of ideas in increments of 5%. This exercise then grows to 3,283 runs (7 thresholds x 469 runs).

Summary of Step 2. On this reduced corpus of ideas, we run our standard out-of-sample LASSO regression from Section 5. For this step, we test all 8 model specifications in Table 4.

Results. Table 6 shows the results of the two-step approach. After extensive testing, pairs of predictors work better than single predictors or three predictors. For pairs, the best results occur when using Word Count and Topic Atypicality and a threshold of 25%. Thus, these two predictors reveal enough information about the content of ideas to screen out those that merely add noise. Prior to this, managers knew that the corpus of ideas contained a lot of junk but did not have a simple way to screen that out. Step 1 does not sacrifice any winners when screening out the bottom ranked 8% of ideas. By comparison, Word Count as a naïve benchmark, sacrifices two winners when screening out the bottom 8% of ideas. Thus, while Word Count is simpler, in this context it performs less well for managers whose cost function for sacrificing winners is steep. In Step 2, Word Colocation performs best of the 8 model specifications in Table 4. Step 2 screens out the bottom ranked 13% of ideas in addition to those screened out in Step 1. Both steps together yield a screening rate of appr. 21% *without sacrificing a single winner*.

The code to run the two-step approach is in Web Appendix 14.

7.2 Relationship of Theory-Based Predictors to Managerial Ratings

Theory-based models are most relevant for managers if they relate to managers' own ratings of ideas. We draw on additional information available in the corpus of ideas' ratings to identify any such relationship. In each contest, experts rated ideas on various dimensions that differ among contests on a 5-point scale from very low to very high. For 11 contests (see Web Appendix 4 for details), we possess information on ideas' ratings on these dimensions. Managers consider three dimensions relevant in six or more contests: innovativeness of idea (643 ideas), communication of idea (483 ideas), and sales potential (641 ideas).

The cells in Table 7 show Pearson correlation coefficients between each of the theory-based models plus Word Count in rows, and ratings of managers in columns, on selected dimensions. Word Colocation correlates highest with innovativeness of an idea (r: -.12, $p<.05$). Word Atypicality correlates highest with the communication of idea (r: -.31, $p<.05$) and sales potential (r: -.19, $p<.05$). Inspiration Redundancy correlates with innovativeness (r: -.08, $p<.05$) and sales potential (r: -.08, $p<.05$). The naïve

Word Count correlates with each dimension [r (innovativeness of idea): -.07, p<.05; r (communication of idea): .15, p<.05; r (sales potential): .10, p<.05]. Overall, selected dimensions of managerial ratings correlate almost twice as highly with theory-based predictors than with naïve Word Count.

7.3 Application to a New Client

A new, large consumer goods client of Hyve provided a new contest of 2947 ideas for evaluation. Internal experts from the company evaluate all ideas in an extensive process and shortlist 54 ideas. From these 54 ideas they select five ideas for funding, which we call winners.

After the internal evaluation, the division head feels that the company could make an even better use of the contest's ideas. The company pays Hyve to re-evaluate all ideas and to select any number of ideas the experts consider to be "good." Hyve's experts shortlist 1125 ideas. Hyve's experts had no knowledge of the ideas selected by the internal experts – so both evaluations are independent.

We apply our AI models to screen ideas for this contest. We test all 8 model specifications from Table 4 out-of-sample. We report the AUC, optimal screening rate, and percentage of ideas screened before sacrificing a winner.

Table 8 shows the results. The AUC for the model with all predictors is .72, like that obtained from the pooled 21 contests. The model specification with only Word Atypicality screens out 61% of all ideas while sacrificing 24% of shortlisted ideas. The same model specification also screens out *62% of all ideas without sacrificing any one of the 5 winners*. Screening out 62% of ideas without losing a winner in this 22nd contest is much higher than is possible for the prior pooled 21 contests (Table 4). This improvement in performance is perhaps because the number of ideas in this 22nd contest is much higher and the percentage of winners much lower than in any prior contest.

Importantly, in comparison to our models, Hyve's experts screen out 62% of all ideas while sacrificing 4 of 5 winners. This application provides preliminary evidence that the proposed model not only does well, but it does better than external experts. If subsequent research supports this finding, this means that innovation managers and scientists soon have a low-cost tool that can replace work which nowadays often outsourced to externals like Amazon MTurk workers – or that cannot be done properly at

all, e.g., due to confidentiality reasons. Because its predictors come from theory and are clearly defined, the results of the model are more transparent and explainable than are ratings of experts.

7.4 Recommendation Scheme for Managers

Figure 2 provides an overview of our recommendations to managers. If managers seek to avoid losing winners in return for lower screening efficiency, they should use the two-step approach (see Web Appendix 14 for code to run the two-step approach). If instead they focus on decreasing effort evaluating numerous ideas, they should use LASSO based on various metrics to generate the ISE curve (see Web Appendices 13 and 15 for code to do the LASSO and the ISE curve respectively). The simplest predictor is Word Count, which has almost the same accuracy as other predictors. It screens out ideas that are short. However, Word Count is easy to game. If ideators know that is the rule, they can write ideas that are long and wordy. If managers are willing to invest a little extra effort for interpretable results, they should use one of two predictors: 1) Word Atypicality, which screens out atypical ideas and keeps inclusive and rich ideas, or 2) Word Colocation, which retains ideas that use novel words. If managers want the best screening to reduce the time and effort of experts, they should implement a model with all 14 predictors. We presented our results to innovation managers, who typically preferred more screening efficiency (Word Atypicality); Web Appendix 12 contains procedure and results.

8. Discussion

Crowdsourcing generates thousands of ideas. The selection of best ideas is costly because of the limited number, objectivity, and attention of experts. Using a dataset of 21 crowdsourcing contests that include 4191 ideas, we test how AI can assist experts in screening ideas.

8.1 Summary of Results and Contribution

This study has three major findings. First, while even the best previously published theory-based models cannot mimic human experts in choosing the best ideas, a simple model using LASSO can efficiently screen out ideas considered bad by experts. In an additional 22nd hold-out contest with internal and external experts, the simple model does better than external experts in predicting the ideas selected by

internal experts. Second, the authors develop an Idea Screening Efficiency curve that trades off the False Negative Rate against the total ideas screened. Managers can choose the desired point on this curve given their loss function. The best model specification can screen out 44% of ideas, while sacrificing only 14% of good ideas. Alternatively, for those unwilling to lose any winners, a novel two-step approach screens out 21% of ideas without sacrificing a single 1st-place-winner. Third, a new predictor, Word Atypicality, is simple and efficient in screening. Theoretically, this predictor screens out atypical ideas and keeps inclusive and rich ideas. These three findings provide methodological, substantive, and managerial contributions respectively to the literature on ideation.

8.2 Questions and Answers

We now answer questions raised by this research.

First, why does AI do better in screening bad ideas than selecting winners? We suggest three possible reasons. One, our contests are blessed with rich data consisting of many long ideas. Word Atypicality works by screening out short, poorly developed ideas with unique words that may not relate to the client's problem. Two, we derive the semantic network based on Google´s search results of the first 50 results pages when typing in the contest topic (Toubia and Netzer 2017). Words from the contest description may have central positions in the semantic network. Thus, typical ideas tend to be more "on topic" than atypical ideas. Three, experts prefer complete, elaborated, and detailed idea descriptions (Dean et al. 2006; Besemer and Treffinger 1981) potentially because details facilitate knowledgeable experts to assess idea quality (Sukhov et al. 2021).

Second, can AI replace humans in ideation? At the current stage of assessing ideas, AI cannot fully or even partially substitute for human experts. However, AI can assist humans, such as the platform's managers, by screening out bad ideas (Level 1 of AI in ideation), thereby reducing humans' cognitive load and helping them focus on the good ideas. The current study is a first step to narrow the ideation funnel early on. Yet, as research advances and as additional models bring in unique and helpful perspectives, research may develop models to select the best ideas (Level 2 of AI in ideation) or even to automatically generate outstanding ideas (Level 3 of AI in ideation).

Third, how does LASSO compare to multi-armed bandits? Prior methods from computer science deal with distinguishing good from bad ideas based on model fitting on massive data. One important example is the use of multi-armed bandits (e.g., Auer 2002; Fiez et al. 2019; Jain and Jamieson 2018; Jamieson and Jain 2018), which deals with efficiently allocating evaluations across ideas (arms) to get the most learning done. While evaluations are easy and cheap in settings in which lay people can form opinions quickly, e.g., (liking of designs), they are costly in settings with scarce experts. In such settings, our approach uses theory to understand the underlying patterns of experts' judgements. In fact, the strength of theory-based predictors is to screen out bad ideas. So, contest managers could first use our theory-based predictors to reduce the number of ideas for experts to judge.

Fourth, why does AI fare much better in the context of chess, face recognition, and even music (e.g., Berger and Packard 2018) than in ideation? Chess has fixed precise rules for movement of pieces, very clear payoffs, and one goal, which enables models to find the best move by elimination of alternatives. Face recognition inputs millions of faces, which lets models identify a few patterns. Music has a finite number of characteristics and patterns that models can easily pick up. In contrast, ideation has a few winners characterized by a vast number of opaque dimensions.

8.3 Limitations and Future Research

This study has limitations that future research may fruitfully address. First, like all such contests, our contests suffer from survival bias: firms do not commercialize ideas that were not shortlisted. So, one never knows their market success. However, using real expert evaluators from companies goes further than prior studies(e.g., Toubia and Netzer 2017; Stephen, Zubcsek, and Goldenberg 2016). Second, there could be a common bias among experts and AI. For example, both might be biased by ideas that are described more clearly, more eloquently, and with love for detail. We cannot tell based on the data at hand. Third, since we use AI-based approaches based on text, complemented by Stephen, Zubcsek, and Goldenberg´s (2016) Inspiration Redundancy, we do not include additional information, for example from images (we exclude design contests), behavioral data, or background information about the ideators, in our analysis. Fourth, during the screening process experts often refine ideas. While the raw idea might be

unattractive per se, the refined ones might be innovative. Our screening approach is performed at the end of the ideation contests. We do not know and can certainly not rule out whether ideas that were screened out by the experts during the screening process could have been refined to make them innovative in later stages of the new product development process. Fifth, in line with some studies from Toubia and Netzer (2017), we assume ideas to be "good" if experts from Hyve´s clients selected them to be presented to a jury of decision makers. As such, our approach mimics the unknown decision criteria these human experts implicitly apply. While using data from various contests shields our results to some degree against idiosyncrasies of a specific dataset, we cannot rule out that our models and the experts might be biased in the same way. Therefore, our models can assist human experts in what they would be doing otherwise themselves, but we cannot make the claim that our models necessarily identify the best or most innovative ideas. Ideally, the best approach to identify the best ideas would be to develop each idea into a new product and to test its success in real markets success. However, in context of thousands of ideas, many of which are poorly developed and or similar to others, this approach is prohibitively expensive and practically infeasible.

References

Abadie, A, and Kasy, M. 2019. "The Risk of Machine Learning." *Review of Economics and Statistics*. 101(5): 743-762.

Allen, BJ, Chandrasekaran, D, and Basuroy, S. 2018. "Design Crowdsourcing: The Impact on New Product Performance of Crowdsourcing Design Solutions from the 'Crowd'." *Journal of Marketing.* 82(2): 106-123.

Amatriain, X, Lathia, N, Pujol, JM, Kwak, H, and Oliver, N. 2009. "The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web." *Proceedings of the 32nd International SCM SIGIR Conference*: 532-539.

Auer, P. 2002. "Using Confidence Bounds For Exploitation-Exploration Trade-Offs." *Journal of Machine Learning Research*. 3: 397-422.

Bayus, BL. 2013. "Crowdsourcing New Product Ideas Over Time: An Analysis of the Dell Ideastorm Community." *Management Science.* 59(1): 226-244.

Berger, J, Humphreys, A, Ludwig, S, Moe, WW, Netzer, O, and Schweidel, DA. 2020. "Uniting the Tribes: Using Text for Marketing Insight." *Journal of Marketing*. 84(1):1-25.

Berger, J, and Packard, G. 2018. "Are Atypical Things More Popular?" *Psychological Science.* 29(7): 1178-1184.

Berlyne, DE. 1970. "Novelty, Complexity, and Hedonic Value." *Perception and Psychophysics*. 8: 279-286.

Besemer, SP, and Treffinger, DJ. 1981. Analysis of Creative Products: Review and Synthesis. *Journal of Creative Behavior.* 15(3): 158-178.

Bilalić, M, McLeod, P, and Gobet, F. 2008. "Inflexibility of Experts—Reality or myth? Quantifying the Einstellung Effect in Chess Masters." *Cognitive Psychology.* 56(2): 73-102.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blei, DM, Ng, AY, and Jordan, MI. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning Research.* 3: 993-1022.

Breiman, L. (2001). Random Forests. *Machine Learning,* 45(1): 5-32.

Brynjolfsson, E, and McAfee, A. 2017. "The business of Artificial Intelligence: what it can – and cannot – do for your organization. *Harvard Business Review* (July 18th) https://hbr.org/2017/07/the-business-of-artificial-intelligence.

Burt, RS. 2004. "Structural Holes and Good Ideas." *American Journal of Sociology.* 110(2): 349-399.

Burt, RS. 2009. *Structural holes: The social structure of competition*. Harvard University Press, Boston, MA.

Cao, W, Li, J, Tao, Y, and Li, Z. 2015. "On Top-k Selection in Multi-Armed Bandits and Hidden Bipartite Graphs." *NIPS.* 8(1): 3-32.

Cockburn, I, Henderson, R, and Stern, S. 2019. "The Impact of Artificial Intelligence on Innovation: An Exploratory Analysis". In Agrawal, A, Gans, J, and Goldfarb, A [eds.]. *The Economics of Artificial Intelligence*. University of Chicago Press, Chicago, IL, 115-148.

Dahan, E, and Hauser, JR. 2002. "Product Development – Managing a Dispersed Process." Weitz, B, and Wensley, R, eds. *Handbook of Marketing.* Sage Publication, New York, NY, 179-222.

Dahl, DW, Chattopadhyay, A, and Gorn, GJ. 1999. "The Use of Visual Mental Imagery in New Product Design." *Journal of Marketing Research.* 36(1): 18-28.

Dahl, DW, and Moreau, CP. 2002. "The Influence and Value of Analogical Thinking During New Product Ideation." *Journal of Marketing Research.* 39(1): 47-60.

Dean, DL, Hender, JM, Rodgers, TL, and Santanen, E. 2006. "Identifying Good Ideas: Constructs and Scales for Idea Evaluation." *Journal of Association for Information Systems*, 7(10): 646-699.

Eling, K, Griffin, A, and Langerak, F. 2014. "Using Intuition in Fuzzy Front-End Decision-Making: A Conceptual Framework." *Journal of Product Innovation Management.* 31(5): 956-972.

Fiez, T, Jain, L, Jamieson, K, and Ratliff, L. 2019. "Sequential Experimental Design for Transductive Linear Bandits," *33rd Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada: 10667-10777.

Friedman, J. H., & Popescu, B. E. (2008). Predictive Learning Via Rule Ensembles. *Annals of Applied Statistics*, *2*(3): 916-954.

Friedman, J, Hastie, T, and Tibshirani, R. 2001. *The elements of statistical learning* (Vol. 1, No. 10). Springer series in statistics, New York, NY.

Füller, J, Hutter, K, Wahl, J, Bilgram, V, and Tekic, R. 2022. How AI revolutionalizes innovation management – Perceptions and implementation preferences of AI-based innovators. *Technological Forecasting & Social Change.* 178: 121598.

Giora, R. 2003. *On Our Mind: Salience, Context, and Figurative Language.* Oxford University Press, New York, NY.

Godart, FC, and Galunic, C. 2019. "Explaining the Popularity of Cultural Elements: Networks, Culture, and the Structural Embeddedness of High Fashion Trends." *Organization Science.* 30(1): 151-168.

Goldenberg, J, Mazursky, S. 2002. *Creativity in Product Innovation.* Cambridge University Press, Cambridge, MA.

Goodfellow, I., Bengio, Y., & Courville, A. 2016. *Deep learning*. MIT press, Cambridge, MA.

Hill, S, and Ready-Campbell, N. 2011. "The Wisdom of (Experts in) Crowds." *International Journal of Electronic Commerce.* 15(3): 73-101.

Hosmer DW, and Lemeshow, S. 2013. *Applied Logistic Regression, 2nd ed*. John Wiley & Sons, New York, NY.

Ireland, ME, and Pennebaker, JW. 2010. "Language Style Matching in Writing: Synchrony in Essays, Correspondence, and Poetry." *Journal of Personality and Social Psychology. 99*(3): 549-571.

Jamieson, K, Jain, L. 2018. "A Bandit Approach to Multiple Testing with False Discovery Control." *32nd Conference on Neural Information Processing Systems (NeurIPS)*. Montréal, Canada: 1-11.

Jamieson, K, Jain, L, Fernandez, C, Glattard, N, Nowak R. 2015. "NEXT: A System for Real-World Development, Evaluation, and Application of Active Learning." *Advances in Neural Information Processing Systems:* 2656-2664.

Jain, L, and Jamieson, K. 2018. "Firing Bandits: Optimizing Crowdfunding." *Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden: 2206-2214.

Jain, L, Mason, B, and Nowak, R. 2017. "Learning Low-Dimensional Metrics." *31st Conference on Neural Information Processing Systems (NIPS)*. Long Beach, CA: 4139-4147.

Katariya, S, Jain, L, Sengupta, N, Evans, J, and Nowak, R. 2018. "Adaptive Sampling for Coarse Ranking." *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*. Lanzarote, Spain: 1-16.

Kilgour, M, Koslow, S, and O'Connor, H. 2020. "Why Do Great Creative Ideas Get Rejected?" *Journal of Advertising Research*. 60(1): 12-27.

King, G, and Zeng, L. 2001. "Logistic Regression in Rare Events Data." *Political Analysis*. 9(2): 137-163.

Kornish, LJ, and Ulrich, KT. 2011. "Opportunity Spaces in Innovation: Empirical Analysis of Large Samples of Ideas." *Management Science.* 57(1): 107-128.

Lamb, C, Brown, DG, and Clarke, C. 2015. "Human Competence In Creativity Evaluation." In Toivonen, H, Colton, S, Cook, M, and Ventura, D, eds. *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*. Brigham Young University, Park City, UT: 102-109.

Luo, L, and Toubia, O. 2015. "Improving Online Idea Generation Platforms and Customizing the Task Structure on the Basis of Consumers' Domain-Specific Knowledge." *Journal of Marketing.* 79(5):100-114.

Matz, SC, and Netzer, O. 2017. "Big Data as a Window into Consumers' Psychology." *Current Opinion in Behavioral Sciences.* 18: 7-12.

Mervis, CV, and Rosch E. 1981. "Categorization of Natural Objects." *Annual Review of Psychology*. 32(1): 89-115.

Mobley, MI, Doares, LM, and Mumford, MD. 1992. "Process Analytic Models of Creative Capacities: Evidence for the Combination and Reorganization Process." *Creativity Research Journal.* 5(2): 125-155.

Netzer, O, Feldman, R, Goldenberg, J, and Fresko, M. 2012. "Mine Your Own Business: Market-Structure Surveillance Through Text Mining." *Marketing Science.* 31(3): 521-543.

Netzer, O, Lemaire, A, and Herzenstein, M. 2019. "When Words Sweat: Written Words Can Predict Loan Default in the Text of Loan Applications." *Journal of Marketing.* 56(6): 1-81.

O'Quin, K, and Besemer, SP. 1999. "Creative products." *Encyclopedia of Creativity*. 1: 413-422.

Rafieian, O, and Yoganarasimhan, H. 2021. "Targeting and Privacy in Mobile Advertising." *Marketing Science*, 40(2): 193-218.

Ritchie, G. 2001. "Assessing Creativity." In Wiggins, GA, ed. *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*. SSAISB, Brighton, UK: 3-11.

Rosch, E, Simpson, C, and Miller, RS. 1976. "Structural Bases of Typicality Effects." *Journal of Experimental Psychology*. 2(4): 491-502.

Runco, MA. 1995. "Insight for Creativity, Expression for Impact." *Creativity Research Journal.* 8(4): 377-390.

Sievert, S, Ross, D, Jain, L, Jamieson, K, Nowak, R, and Mankoff, R. 2017. "NEXT: A System to Easily Connect Crowdsourcing Adaptive Data Collection." *Proceedings of the 16th Python in Science Conference* (SCIPY). Austin, TX: 113-119.

Simonton, BK. 1999. "Creativity from a Historiometric Perspective." In Sternberg, RJ, ed. *Handbook of Creativity*. Cambridge University Press, Cambridge, UK: 116-136.

Stephen, A, Zubcsek, PP, and Goldenberg, J. 2016. "Lower Connectivity Is Better: The Effects of Network Structure on Redundancy of Ideas and Customer Innovativeness in Interdependent Ideation Tasks." *Journal of Marketing Research.* 53(2): 263-279.

Sukhov, A. 2018. "The Role of Perceived Comprehension in Idea Evaluation." *Creativity and Innovation Management.* 27(2): 183-195.

Sukhov, A, Sihvonen, A, Netz, J, Magnusson, P, and Olsson L. 2021. "How Experts Screen Ideas: The Complex Interplay of Intuition, Analysis, and Sensemaking." *Journal of Product Innovation Management*. 38(2): 248-270.

Terwiesch, C, and Ulrich, KT. 2009. *Creating and Selecting Exceptional Opportunities.* Harvard University Press, Cambridge, MA.

Tibshirani, R. 1996. “Regression Shrinkage and Selection Via the Lasso.” *Journal of the Royal Statistical Society,* Series B (Methodological). 58(1): 267–288.

Toubia, O. 2006. “Idea Generation, Creativity, and Incentives.” *Marketing Science.* 25(5): 411-425.

Toubia, O, and Flores, L. 2007. “Adaptive Idea Screening Using Consumers.” *Marketing Science.* 26(3): 342-360.

Toubia, O, and Netzer, O. 2017. “Idea Generation, Creativity, and Prototypicality.” *Marketing Science.* 36(1): 1-20.

Urban, GL, and Katz, GM. 1983. “Pre-test-market models: Validation and managerial implications.” *Journal of Marketing Research*. 20(3): 221-234.

Wang, M, Zhao, Y, and Zhang, B. 2015. “Collective Dynamics of ‘Small-World’ Networks.” *Scientific Reports*. 5(1): 1-12.

Ward, TB. 1995. “What’s Old About New Ideas?” In Smith, SM, Ward, TB, and Fiske, RA, eds. *The Creative Cognition Approach.* MIT Press, Cambridge, MA*:* 157-178.

Watts, DJ, and Strogatz, SH. 1998. “Collective Dynamics of ‘Small-World’ Networks.” *Nature*. 393(6684): 440-442.

Wedel, M, and Kannan, PK. 2016. “Marketing Analytics for Data-Rich Environments.” *Journal of Marketing.* 80(6): 97-121.

Wessling, KS, Huber, J, and Netzer, O. 2017. “MTurk Character Misrepresentation: Assessment and Solutions.” *Journal of Consumer Research*. 44(1): 211-230.

Wooten, JO, and Ulrich, KT. 2019. “The Impact of Visibility in Innovation Tournaments: Evidence from Field Experiments.” *Working Paper*: 1-36.

Table 1. Intuition of Original and Extended Models

	Intuition of Original Approach	Original Approach	Intuition of Extension	Extension
Word Colocation (Toubia, Netzer 2017)	Semantic Network Approach: Good ideas are prototypical, i.e., they balance novelty and familiarity (represented by Google search results)	Reference Corpus 1. Enter contest title and description in **Google Search** 2. Take first 50 pages and read html code of these pages 3. Screen out stopwords from code (e.g., "the" and "and") 4. Lemmatize the text (only words remain) 5. Build semantic network for each contest a. Nodes: number of pages, on which respective word was used b. Edges: number of pages, on which nodes connected by the edge occur jointly Idea 1. Idea's semantic network uses idea's words and edges from the reference corpus 2. Calculation of metrics from Table 2 3. Toubia and Netzer's (2017) key metric Prototypicality: a. computes ECDFs for each document of reference corpus b. takes the average of those results c. compares ECDF of given idea to average using KS distance (KS distance = maximum absolute difference between both vectors)		
Topic Atypicality (Berger, Packard 2018)	Semantic Network Approach: Ideas are better if they are different from other ideas submitted to the contest	Implemented analogously to Berger and Packard 1. LDA generates a topic model from corpus of ideas 2. LDA model output: set of linear equations, one for each topic, equation terms represent words; word terms' coefficients indicate importance of word to topic. 3. Topic Atypicality $A_{TA}(C,i)$: distance between idea and location of overall corpus in topic space. For topic *l,* idea *i,* reference corpus *C* $$A_{TA}(C,l,i) = \frac{1}{1 - \ \lvert l(C) - \ l(i)\rvert / (l(C) + \ l(i) + .001)}$$ Total Topic Atypicality, $A_{TA}(C,i)$ is calculated by taking the average of each $A_{TA}(C,l,i)$ over topics *l*.	LDA extracts popular (common) topics (dimensions), e.g., word bundles. LDA may miss unique words as "errors"; successful new product ideas tend to be novel or unique. Metrics that capture Word Atypicality may be superior.	1 - Jaccard distance between word sets; W_C = set of words in C; W_i= set of words in *i*; Word Atypicality: $$A_{WA}(C,i) = 1 - \frac{W_C \cap W_i}{W_C \cup W_i}$$

Inspiration Redundancy (Stephen, Zubcsek, Goldenberg 2016)	Social Network Approach: Ideators with access to diverse information, i.e., with contacts that don't talk with each other, submit better ideas	a) Original Version of Clustering Coefficient Metric calculated on undirected network of comments of ideator *n* at end of the contest. - advantage: makes use of all information at the end of the contest to evaluate ideas - Clustering Coefficient for node *n* in an undirected graph: number of edges among n's neighbors that exist at the end of the contest divided by the number that could exist. - if N_n is "neighborhood" of *n* at the end of the contest, *m* and *p* are neighbors of *n*, e_{mp} is an edge between *m* and *p*, v_m and v_p are nodes for *m* and *p,* and node *n* has k_n total neighbors, the Clustering Coefficient CC_n is: $$CC(n, N_n) = \frac{2|\{e_{mp}: v_m, v_p \in N_n\}|}{k_n(k_n - 1)}$$ b) Modified Version of Clustering Coefficient Metric calculated on undirected network of comments of ideator at the point of idea submission. - advantage: only considers information available at point of idea submission: no endogeneity - Clustering Coefficient for node *n* in an undirected graph at time *t*: number of edges among n's neighbors that exist at the time of idea submission divided by the number that could exist. - if $N_n(t)$ is "neighborhood" of *n* at time *t, m* and *p* are neighbors of *n,* e_{mp} is an edge between *m* and *p*, v_m and v_p are nodes for *m* and *p,* and node *n* has $k_n(t)$ total neighbors at time *t*, the modified Clustering Coefficient CC_n at time *t* is: $$MCC(n, N_n(t), t) = \frac{2|\{e_{mp}: v_m, v_p \in N_n(t)\}|}{k_n(t)(k_n(t) - 1)}$$	SZG's metric for Inspiration Redundancy only considers network structure of 1st degree contacts; operationalization by Burt (2004) allows to consider 2nd degree contacts	- a_{nm} indicates existence of link between *n* and *m* (1 if link present, 0 otherwise) - $p_{nm}(t) = a_{nm}/N_n(t)$; $N_n(t)$ is neighbourhood of *n* at time *t* - Constraint metric $C(i, N_n(t), t)$= $\sum_{m \neq n}(\rho_{nm}(t) + \sum_{p \neq n; p \neq m} \rho_{np}(t)\rho_{pm}(t))^2$ - measure sums across neighbors *m* of a node *n*

Table 2. Comprehensive List of Predictor Variables in Various Models

Source	Variables	Computed from which data?
TN	Mean, min, max, and coef. of variation of Jaccard indices between word pairs	Google Search
TN	Mean, min, max, and coef. of variation of node frequencies	Google Search
TN	Kolmogorov-Smirnov distance from Toubia and Netzer (2017)	Google Search
BP	Word Atypicality	All ideas from the same contest
BP	Topic Atypicality	All ideas from the same contest
SZG	Degree	Comments network
SZG	Modified Clustering Coefficient (a.k.a. Transitivity)	Comments network
SZG	Constraints (Burt's metric)	Comments network

TN = Toubia and Netzer, SZG = Stephen, Zubcsek and Goldenberg, BP = Berger and Packard

Table 3. Variables Retained by LASSO (in-sample),
Input: Variables inspired by Original Models. DV: Shortlisted (yes/no)

Source	Variable Name	Standardized Coefficient
	Intercept	-3.50
SZG	Original Clustering Coefficient	-0.19
TN	Kolmogorov-Smirnov (Google)	-0.08
BP	Topic Atypicality	-0.07
Control	Percent Shortlisted (contest-level)	0.10

TN = Toubia and Netzer, SZG = Stephen, Zubcsek and Goldenberg, BP = Berger and Packard

Table 4: Out-of-Sample: Eight Model Specifications

Dependent Variable:	*Shortlist*	*Winner*
Predictors in Various Models	**Optimal Screening Rate / % of Good Ideas Screened Out at Optimal Screening Rate**	**Percent Screened Before Losing Winner***
Word Colocation only (TN original)	28%/08%	15%
Topic Atypicality only (BP original)	29%/11%	12%
Inspiration Redundancy only (SZG original)	24%/09%	10%
Word Atypicality only (this study)	40%/13%	12%
Word Atypicality (Google)	40%/14%	15%
Word Count only (naïve)	40%/15%	13%
Model with Three Theoretical Predictors (Word Colocation, Topic Atypicality, Inspiration Redundancy)	35%/13%	14%
Model with all Predictors (see Table 2)	44%/14%	12%

* after accounting for percentage of winners in contests
**AUC of all Models >.7; even small differences in AUC translate to substantial differences in screening rate, which is the managers' prime concern

Table 5: Overlap Between Model's Predictions of Top 25% (brown) and Screening of Bottom 25% (blue); expected overlap by chance: 6.25%

	Top 25% of Ideas Predicted by Each Model / Bottom 25% of Ideas Predicted by Each Model			
	1. Word Colocation	2. Topic Atypicality	3. Inspiration Redundancy	4. Word Atypicality
1. Word Colocation				
2. Topic Atypicality	7.4%* / 6.6%			
3. Inspiration Redundancy	5.4% / 7.4%*	5.3% / 5.4%		
4. Word Atypicality	**10.4%*/12.6%***	**9.3%*/ 7.5%***	**7.4%*/ 8.0%***	

*p<.05; **bold**: both metrics significantly overlap at the top AND at the bottom

Table 6: Best Performing Model in Two-Step Analysis:
First Step: Screen out Ideas in Bottom 25% According to Both Predictors
Second Step: LASSO with Predictors Listed in Second Column

First Step (Best of 105 Possible Pairs of first five Models from Table 5)	Second Step (Best of All 8 Models from Table 5)	Percent Screened Before Losing Winner
Topic Atypicality and Word Count	Word Colocation	21%

Table 7. Managerial Relevance of the Theory's Metrics

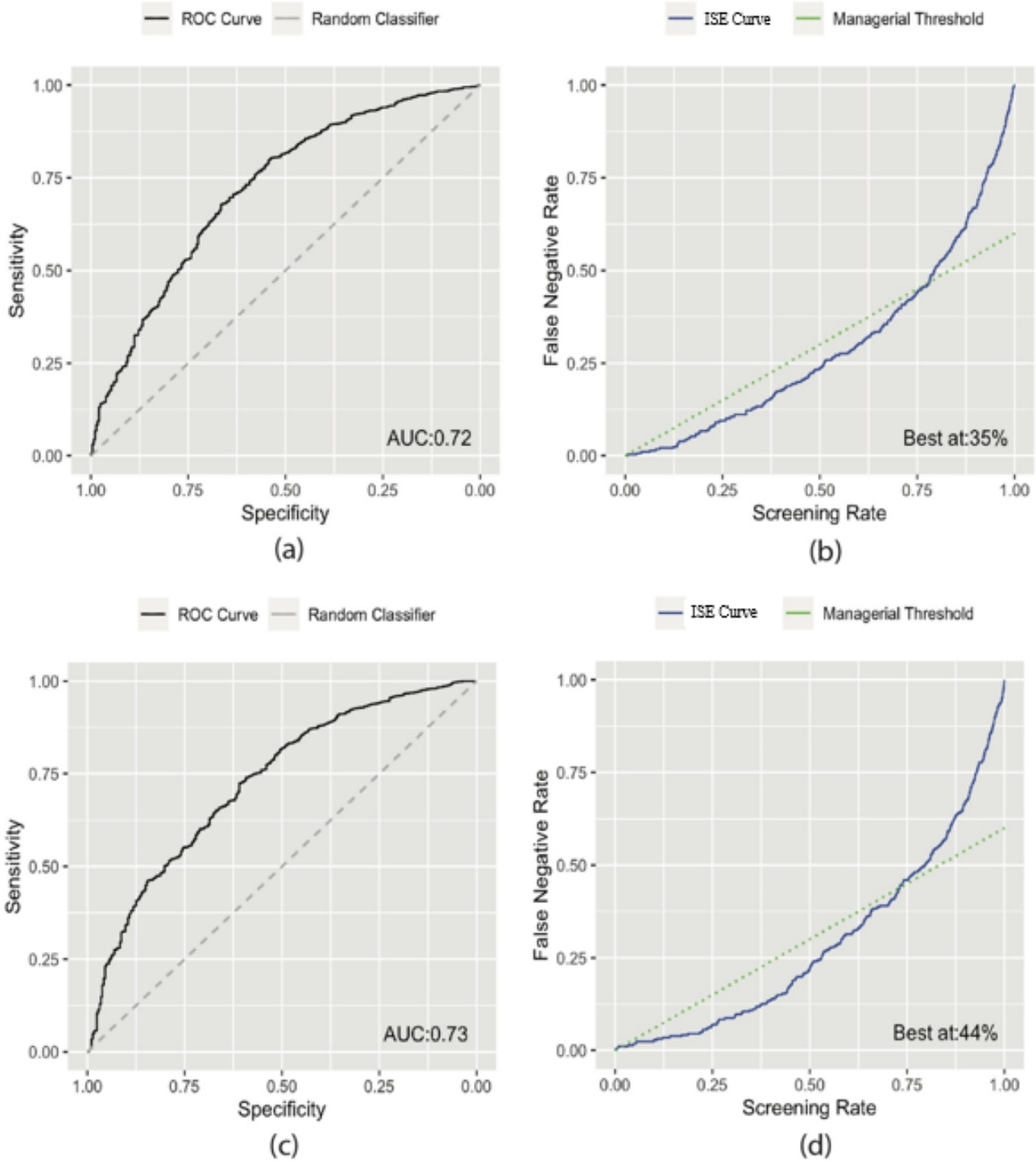
		Managerial Dimensions		
Source Theory	Metric	Innovativeness of Idea (643 ideas)	Communication Quality of Idea (483 ideas)	Sales Potential (641 ideas)
		Pearson Correlation Coefficients		
Word Colocation (TN)	Kolmogorov-Smirnov	-.12*	-.22*	-.04
Topic Atypicality (BP)	Peer Deviation LDA	.03	-.03	.02
Word Atypicality	Peer Deviation Jaccard	-.07*	-.31*	-.19*
Inspiration Redundancy (SZG)	Clustering Coefficient	-.08*	.03	-.08*
Naïve Model	Word Count	.07*	.15*	.10*

p<.05; green: best-performing metric of the two important idea-dimensions.

Table 8. Out-of-Sample: Predictors Included in Models: Internal Contest

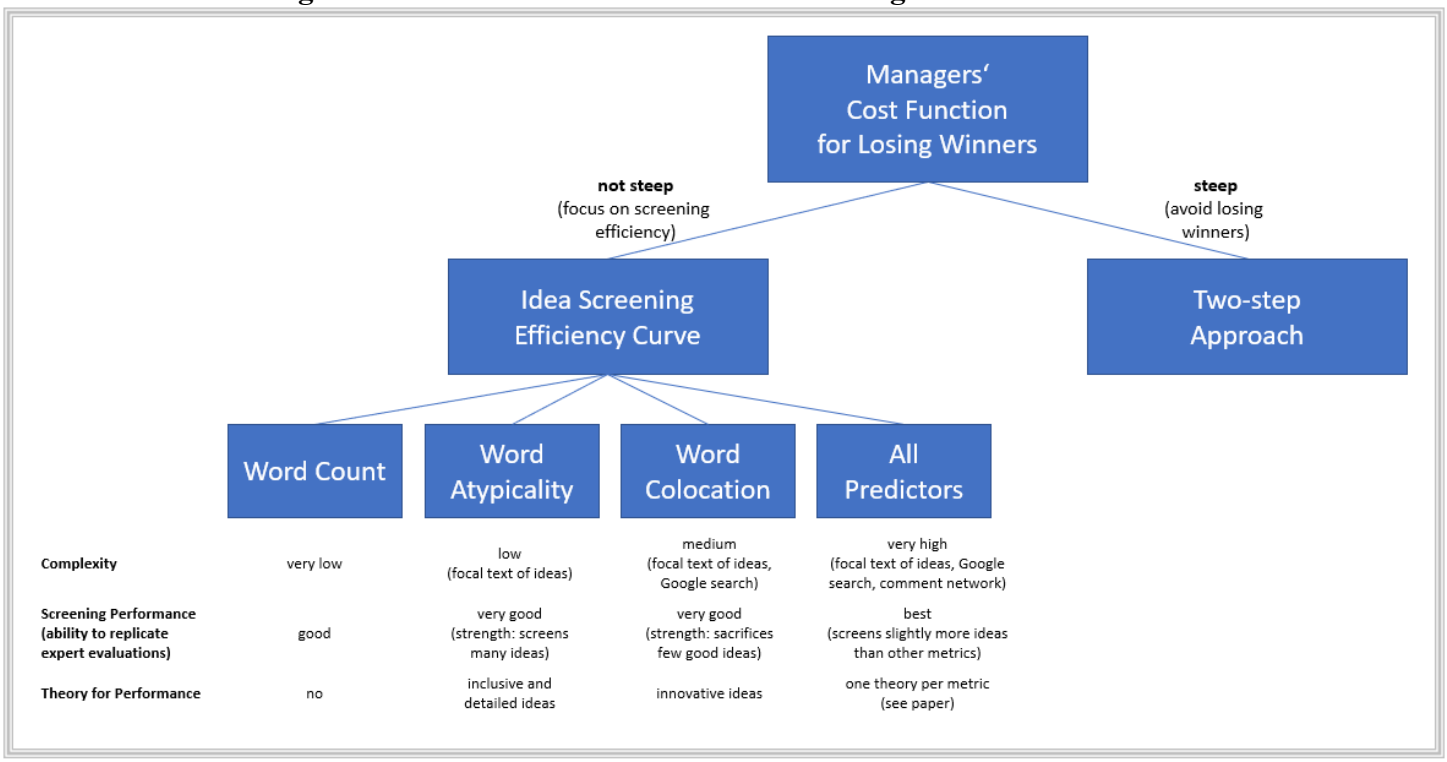
Dependent Variable:	*Shortlist*	*Winner*
Models	**Optimal Screening Rate / % of Good Ideas Screened Out at Optimal Screening Rate**	**Percent Screened Before Sacrificing a Winner***
Word Colocation only (TN original)	37% / 15%	44%
Topic Atypicality only (BP original)	11% / 4%	5%
Inspiration Redundancy only (SZG original)	Unavailable	-
Word Atypicality only (this study)	61% / 24%	62%
Word Atypicality (Google corpus)	59% / 22%	69%
Word Count only (naïve)	58% / 24%	67%
Model with Three Theoretical Predictors (Word Colocation, Topic Atypicality, Inspiration Redundancy)	11% / 4%	8%
Model with all Predictors (see Table 2)	61% / 24%	62%

Figure 1. Model with three theoretical predictors: ROC Curve (a),
Model with three theoretical predictors: Idea Screening Efficiency Curve (b),
Model with all predictors from Table 2: ROC Curve (c),
Model with all predictors from Table 2: Idea Screening Efficiency Curve (d)



Sensitivity = True Positives / All Positives
Specificity = True Negatives / All Negatives
False Negative Rate = False Negatives / All Positives = 1 - Sensitivity
Screening rate = Number of Ideas Dropped / Number of All Ideas

Figure 2. Recommendation Scheme for Managers to Screen Ideas



Web Appendix 1: Multi-Armed Bandits

Bandits are a commonly used algorithm in industry and computer science for selecting a solution from a set of candidates. Indeed, some ideation contests tap simple alternatives (e.g., captions) with many raters (like the New Yorker Caption Contest) (Tanczos et al., 2017).

Bandits optimally sample payoffs under uncertainty. The original context is a gambler sampling a group of machines by pulling arms to discover which gives the best payoff. In our case, the arms are ideas, the act of pulling an arm is soliciting an evaluation of an idea and the payoffs are evaluations. The uncertainty in the bandit problem comes from the fact that the payoffs at each arm vary according to a distribution whose parameter values are unknown. Bandits cover a variety of use cases, but all have in common sequential sampling from the arms.

Importantly, there are variations in the bandit problem that lead to different methods. For the present work, the most important distinction is whether the "gambler" hopes to maximize cumulative rewards (which requires balancing exploration vs. exploitation) or only information about the reward distributions of the arms (when exploitation is of no value). The former is the classical bandit problem, the latter is referred to as a "pure exploration" bandit (e.g., the *New Yorker* Caption Contest (Sievert et al. 2017), where the goal is to find the funniest text to accompany a cartoon).

The goal of a pure exploration bandit is to identify the best arm with a fixed degree of confidence using as few samples as possible.[4] The question of when to use a pure exploration bandit as opposed to a bandit which tries to both, explore and exploit, depends on whether the rewards *during* sampling are of any value beyond the information they provide about arms. Evaluations of an idea from a rater are not of direct value, they are only useful to determine which ideas to screen out, so the pure exploration bandit is the appropriate type for our case.

For some bandit use-cases, the task is to determine the best arm and is referred to as "best arm identification." In our case, we need to learn about the group of best arms (shortlisted ideas). This is called the "top-k" arm identification task, where k is the size of the group of best arms.

To compare our theory-based predictive model with the performance of bandits, we randomly select two contests. The first contest is the RLB contest, which aims to find solutions that already exist in the B2C market that can be applied to the B2B market; 70 ideas enter the contest. The second contest is the Allgau2 contest, which aims to obtain application-oriented solutions and best practices for cost-effective retrofits of charging infrastructure in underground car

[4] Sometimes the goal is the inverse: to identify the best arm with as much confidence as possible given a fixed number of samples.

parks; 38 ideas enter this contest. To implement bandits, for each contest we solicit new evaluations of the ideas from two groups each of 24 students who participate for course credit. Each group of students participates in evaluating only one contest. There is no overlap between the groups. Each student evaluated every idea from the assigned contest. Students report evaluation times between one and three days. These evaluations represent a cumulative evaluation effort of hundreds of hours, which on the free market is extremely costly

We adopt the "top-k, pure exploration" bandit (Cao et al., 2015). Web Appendix 2 shows that appropriate application of bandits in our setting, at the very minimum, requires a huge number of experts' evaluations of ideas. Thus, reaching the required minimum threshold for the applicability of bandits seems extremely challenging. We contacted the author of a well-known bandit paper, Wei Cao (Cao et al. 2015). According to Wei Cao, for smaller datasets, the "trivial" approach of taking the average rating for each idea might be better (Personal communication, January 13th, 2021).

Table WA1.1 shows the AUC for Word Atypicality on RLB and Allgau2 contests, along with the AUC when average ratings from the students are used. The score is marginally higher for the RLB contest, and, strangely, inversely related for the Allgau2 contest (AUC of 0.37 indicates worse than random.) Thus, to summarize:

1) Bandits, particularly top-k bandits that identify sets of ideas, are built for huge samples and typically use many evaluators. Evaluations tend to be rather simple and evaluators usually perform many evaluations in an acceptable amount of time. In contrast, our setting is characterized by non-trivial ideas with long descriptions, i.e., a high information setting. Only a few experts can reliably judge the quality of these ideas; they are scarce and costly. Thus, bandits are not efficient in this specific context.

2) Even if there were enough experts, bandits may require a prohibitively costly number of ratings to add value. In our setting, in total, students invested several hundred hours of work – yet this information was not enough for bandits to obtain reliable results. To obtain such evaluations from industry experts may be (prohibitively) expensive.

3) The comparison between our theory-based predictive model and top k bandits shows that it is the setting which determines which of the two approaches predicts better. Bandits are likely to outperform our theory-based model in settings where many evaluators can quickly provide additional responses without much additional investigation, i.e., in lower information settings, such as when customers are asked for their personal preferences. Yet, theory-based models may be more applicable in high-information settings in which there are few experts and evaluations may take a lot of

time. The advantages of a theory-based predictive model are that it: a) is instantaneous, b) is free of cost, c) is free of confidentiality concerns[5], d) has substantive insights, and e) has lower needs for data.

4) The multi-armed bandits and theory-based models may complement each other. Where our theory-based model manages to screen bad ideas, bandits typically do a good job in selecting the winner(s). So particularly in settings with a limited number of raters, contest managers could first use our theory-based model to reduce the number of ideas that need to be evaluated, and then collect the evaluations needed to run the multi-armed bandits. Thereby, our theory-based model can even enhance the applicability of bandits.

Table WA1.1: Bandit Results

	RLB	Allgau2
Ranked by Mean	0.73	0.37
Word Atypicality	0.72	0.55

[5] We thank the editor for this succinct statement of the benefits.

Web Appendix 2: Bandit Data Requirements

Cao et al. (2015) provide a lower bound on "sample complexity" for top-k pure exploration bandits. The sample complexity relates negatively to the amount of imprecision and uncertainty we accept.

Our goal is to show that bandits are not appropriate for our setting. We do this by choosing the contest and the parameter values so they are most favorable for bandits, i.e., we will accept the *maximum imprecision and uncertainty for which the mathematical result in Cao et al. (2015) still holds*. Cao et al. (2015) provide a proof of their lower bound on sample complexity that holds under the following conditions: the tolerance parameter, ϵ, can be at maximum ¼, and the probability of error, δ, can be at maximum 1/48. For the sake of our illustration, we use those maximum values, as they represent the highest tolerance for inaccuracy and the highest acceptance of probability of error. Finally, we use the lower bound instead of the upper bound on sample complexity. The formula for sample complexity helps to judge the feasibility of collecting enough data to apply bandits. The lower bound formula is:

$$\Omega\left(\frac{n}{\epsilon^2}\frac{1}{\theta_{avg}(B)}\left(1+\frac{\log\frac{1}{\delta}}{k}\right)\right) \qquad A1$$

where n is the number of arms (ideas), B is the set of all arms (ideas), and k is the number of arms to be selected as "best" (number of shortlisted ideas). The Ω indicates that this bound holds as n gets very large.[6] This suggests that bandits are designed for settings with much larger data sets or much lower costs of data collection than we have. Many contests have very few ideas (sometimes fewer than 50). In practice, for low ratios of n/k, this lower bound is likely to be an extremely poor approximation, making the answer provided by bandits unreliable.

Even so, the formula can give a general idea of the order of magnitude of samples needed. For the smaller of the two contests, Allgau2, the term inside the parentheses is:

$$\frac{39}{0.0625}\frac{1}{\theta_{avg}(B)}\left(1+\frac{\log 48}{3}\right)\approx 973\frac{1}{\theta_{avg}(B)} \qquad A2$$

The term $\theta_{avg}(B)$ is an average of the mean of the reward distributions in B, the entire set of arms. In this setting, the rewards are scaled to the interval $(0,1]$, so 973 is the lowest value this formula could yield for the number of samples (idea evaluations) required to use bandits on the Allgau2 contest. In realistic scenarios it could easily be much higher, by several orders of magnitude. For instance, even a small increase in the tolerance parameter to $\epsilon = 1/10$ yields

[6] The use of Ω is a convention in asymptotic notation to designate lower bounds.

a sample complexity of 6,048, and if $\theta_{avg}(B)$ were nearer to a more reasonable 0.5, sample complexity increases by a factor of two, to 12,096. Allgau2 is also the smallest contest. Since the sample complexity grows with the number of ideas (n), if top-k pure exploration bandits require too many evaluations even for Allgau2, larger contests will be even less tractable for bandits.

We have 912 evaluations. These evaluations represent a cumulative evaluation effort of hundreds of hours, which on the free market is extremely costly. Even for the smallest contest by far, under the most favorable conditions, i.e., where we accept the maximum imprecision and uncertainty for which the mathematical result in Cao et al. (2015) still holds, we were not able to estimate the bandit. For contests with more submissions, it will be much more difficult to find a suitable number of experts.

Thus, reaching the required minimum threshold for the applicability of bandits seems extremely challenging.

Web Appendix 3: How Word Atypicality Works

- Case 1, idea *i* uses only words that other ideas also use:
  - $W_i = a, b, c$
  - $W_c = a, b, c$
  - $Measure = 1 - (a + b + c)/(a + b + c) = 0$

- Case 2, idea *i* uses one word in common with W_c
  - $W_i = a, b, c$
  - $W_c = c, d, e$
  - $Measure = 1 - c/(a + b + c + d + e) = 1 - 1/5 = 0.8$

- Case 3, only 1 word does *not* overlap
  - $Wi = a, b, c$
  - $Wc = b, c, d$
  - $Measure = 1 - (b + c)/(a + b + c + d) = 1 - 2/4 = 0.5$

Web Appendix 4: Contests and their Descriptions

Contest	Description	Vocabulary Size After Cleaning and Lemmatization
Allgau1*	Bicycle trailers are often wide and bulky, which can be annoying for others living in an apartment building as many residents do not have a garage to store them. We want to build a trailer for bikes and scooters. The trailer shall be environmentally friendly and save space. Boxes of various sizes as well as the transport frame shall be developed. A combination bike trailer in a modular form would make the project particularly exciting. We would like to use bio-composite building materials. The boxes and the trailer shall be attractive, strong, durable, and fluid-resistant (leaking waste). Furthermore, they shall be multi-functional, enabling you not only to transport your scrap to the recycling center or your purchases from the weekly market, but perhaps also to carry your children. CO2 neutral production would be fantastic but it is not mandatory. The trailers and the boxes shall be offered at an affordable price.	2620
Allgau2*	The aim is to obtain application-oriented solutions and best practices for cost-effective retrofits of charging infrastructure in underground car parks with a focus of cost minimizing retrofit solutions and without an increase of the power supply value.	1187
Deloitte and Telekom Ideabird	Ideas about potential machine-to-machine (M2M) application areas and solutions in finding and following a variety of objects or any other entity.	6026
DHL City Open Innovation	Identify cities' logistics needs through today's technological trends and develop solutions that will address tomorrow's challenges. Three categories: city efficiency, green city, digital logistics for one of the world's leading logistics companies.	1145
Fraport Airport of the Future	Frankfurt airport launched this Innovation Challenge as an opportunity for ideators to contribute their own ideas for making Frankfurt Airport an even better, more unique place. They should help the airport to create a fascinating new world: the airport of the future.	4777
Fujitsu PalmSecure Innovation Contest	Current occurrences and increasing security issues worldwide clearly show the indispensable demand for better security wherever IT is used. To moderate these concerns, FUJITSU provides PalmSecure™, a biometric authentication technology that detects the unique pattern of the flowing blood in the palm of your hand with infrared light. The technology persuades due to its reliability and simple, contactless functionality. In cooperation with FUJITSU, the HYVE AG launched an Idea contest with the aim to find new fields of application for the biometric authentication technology.	4260
LedsArt1*	An upscale Shopping Mall packed with luxurious brands, open for out of the box ideas and striving to surprise shoppers with the latest technologies.	1001
LedsArt2*	Christmas is the most important time of the year for shopping malls and retailers. However, to attract many customers, those companies must differentiate themselves from the competition. We want to specialize on large Christmas tree concepts. Our strong manufacturing partner is in need for fresh concepts that he can later build and sell to above mentioned customers. The focus of the challenge is to create an immersive experience inside the tree.	1320

Lufthansa Cargo Air Innovation Challenge 1	Green solutions: Which green processes and solutions along our supply chain can you imagine to be realized in short-, medium-, and long-term horizon? Imagine airfreight in 2020! What will define a green airfreight carrier? Which new innovative services would you expect from a cargo airline?	3903
Lufthansa Cargo Air Innovation Challenge 2	New and creative ideas about how customer service should look and function in the coming years. Focus on customer touch points, where the customer and the customer service department have specific contact; apps / all means of new technology communication and customer loyalty programs.	2575
Mastercard	Mastercard searches for the next big thing in payments technology.	2322
Moskiteiros*	We need new, innovative, and sustainable ways to fight mosquitoes close to where people live. Mosquitoes, especially the Aedes aegypti, carry diseases and cause a lot of human suffering.	1840
MT Aero-space*	MT Aerospace developed a manufacturing technology for geometric medium complex and thick-walled components like pressure vessels, cylinder and spheres out of CFRP (carbon fiber reinforced plastic). This technology is a combination of dry-wrapping/-layering of carbon fibers followed by a vacuum resin infusion. Show, where this manufacturing technology can be used.	2531
Reifenhäuser	Company searches for the best ideas for products, processes, and machines in the context of plastics extrusion.	988
RLB*	The department for corporate clients of RLB is challenging itself in how it interacts with its customers and which services it offers to them. What solutions exist in the B2C (business to consumer) market that could be transferred to the B2B (corporate client) world of the bank?	1923
SEAS*	SEAS-NVE is one of the largest energy providers in Denmark. Besides providing electricity, they offer energy consulting as a service to help companies reduce their energy spending. With businesses becoming digitalized, SEAS-NVE wonders how this will affect and serve them in the future? How would you conceive the future of energy consulting? Join us in reinventing SEAS-NVE's business model!	2989
Vodafone Connected X Challenge	How much time have you spent looking for a free parking slot? How many times have you complained because a full garbage container was standing in front of your house? And what about the fear of losing your pet if you are taking a walk? Imagine all these problems could be solved by using a new way of connection that requires small quantities of data, over a long period of time and even in places that are really hard to reach. Be part of our global innovation community and generate use cases and innovations in Narrowband IoT which will be striking in the future!	5536
Voestalpine*	Voestalpine has developed a process, where conductive structures can be embedded in the paint layer of the steel surface. This new technology makes it possible, for instance, to heat the steel surface or to integrate sensor, control, and monitoring elements into the coating. This new development transforms steel to a "smart product" that still can be reshaped.	3312
Volkswagen*	The car manufacturer Volkswagen is constantly working on the development of new production methods and processes. Due to the complexity of joining different materials and components, VW is looking for new concepts for a body with lower joining complexity that can be mass-produced. How does such a body look like? Which new joining processes can be used? How can classic joining processes be reduced?	2083
Wiesn*	What kind of merchandise would you like to purchase at the Oktoberfest? The Oktoberfest, traditionally called 'Wiesn', is the largest German festival. Next to	1701

	beer and food, all kinds of merchandise can be bought. The aim of this contest is to identify creative and innovative products for Wiesn merchandise. Let your imagination and creativity run free, inspire us, and win a trip to the Wiesn!	
Zeiss VR One App Contest	We are looking for innovative, interesting, and imaginative ideas for apps and complete applications for the ZEISS VR ONE headset. The mobile and system-independent headset provides users with a virtual reality experience via the apps installed on their smartphone. What experiences would you like to have while you are part of this other world?	4832

*we have information on additional dimensions used to evaluate the contests

Web Appendix 5: LASSO

LASSO is a regularized version of regression (Tibshirani, 1996, Sun et al., 2019). Regularization introduces an explicit bias into a predictive model to obtain some desirable property. In the case of LASSO, the method introduces a bias to achieve parsimony: during estimation, the method attempts to assign weights of zero to coefficients of variables with low values. To do so, LASSO uses a penalty parameter, λ. High values of λ lead to fewer predictors. We estimate LASSO using penalized maximum likelihood estimation. Denote by $\hat{\beta}_0$ the estimate of the intercept term and $\hat{\beta}$ the vector of coefficients, not including the intercept. Denote by x_i' row i of the matrix of predictors X, where X is of dimension $N \times P$, where N is the number of observations (ideas) and P is the number of predictors. Then the objective function for a logistic LASSO is

$$(\hat{\beta}_0, \hat{\beta}) = argmax_\beta \; l(\beta_0, \beta) - \lambda \left|\left|\beta\right|\right|_1 \qquad A3$$

where

$$l(\beta_0, \beta) = \frac{1}{N} \sum_{i=1} \left[y_i (\beta_0 + x_i' \beta) - \log\left(1 + e^{\beta_0 + x_i' \beta}\right)\right] \qquad A4$$

where $\left|\left|x\right|\right|_1$ is the L1 vector norm (the sum of the absolute values of the elements of the vector).

To estimate LASSO, a value for λ is required. We choose λ using k-fold cross-validation (Friedman et al., 2001; Sood, Tellis, James 2009). This procedure is explained in detail in Web Appendix 6. We also apply k-fold cross validation (Friedman et al., 2001; Sood, Tellis, James 2009) to measure the accuracy of the model specification is as follows: we hold-out one contest at a time, estimate LASSO on the remaining 20 contests, then predict the held-out contest.

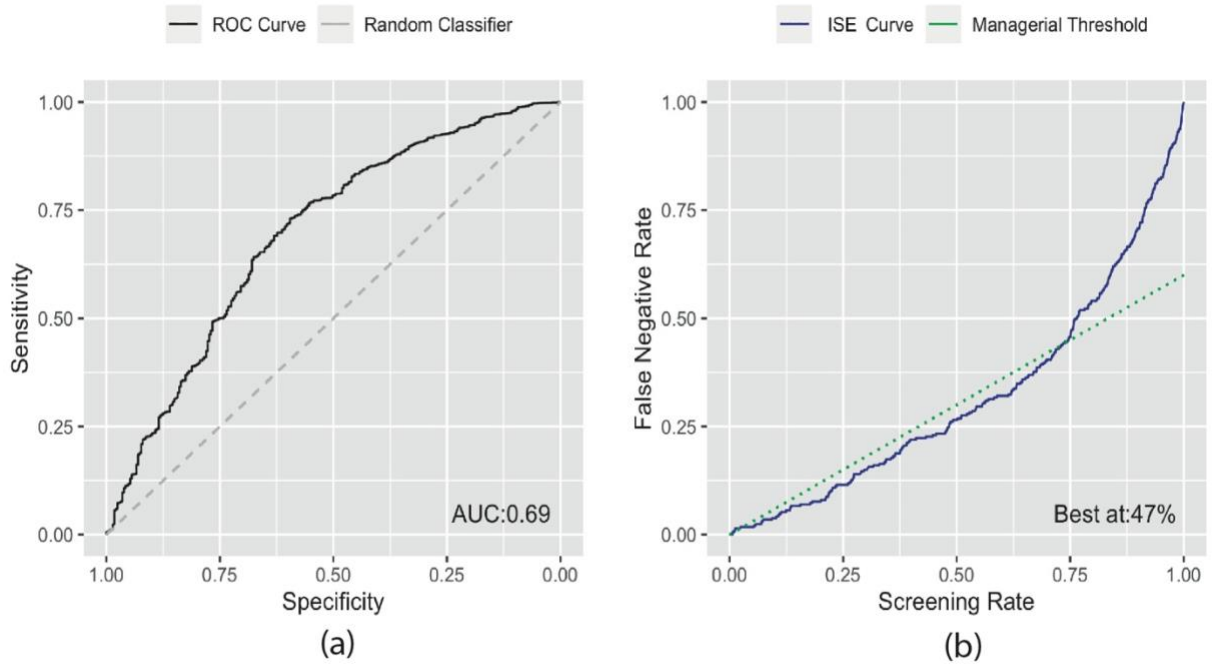
Web Appendix 6: Data Processing and Modeling Steps

1. Tokenization, stop-word removal and lemmatization of all text sources (Google results, idea descriptions)
2. Construction of two different variants of the clustering coefficient: thereby, the original clustering coefficient includes all comments, the modified clustering coefficient includes only outdegree and comments up to idea submission.
3. Define the set of all 21 contests as $Contests$. Define the number of ideas in $contest$ as $nid_{contest}$. Define the number of shortlisted ideas in $contest$ as $nshort_{contest}$. For each contest $contest \in Contests$:
    a. Remove $contest$ from dataset, leaving 20 contests.
    b. Use cross-validation on remaining data to select values for tuning parameters such as λ (parsimony parameter in LASSO). For each $\lambda_t \in (\lambda_{min}, \dots, \lambda_{max})$
        i. Given λ_t, obtain cross validation AUC.
        ii. Cross validation leaves one of the 20 contests out at a time and trains on the remaining 19. Cross validated AUC is the average AUC for λ_t over 20 cross validation steps (1 contest held out per step).
    c. With best-performing values of tuning parameters (λ^*_{CV}) from cross validation, make predictions for $contest$ in the form of probabilities of being shortlisted. Note that since different contests have different proportions of ideas shortlisted, we include a control variable: $\frac{number\ of\ shortlisted\ ideas\ in\ contest}{number\ of\ ideas\ in\ contest} = \frac{nshort_{contest}}{nid_{contest}}$. This variable is excluded from LASSO regularization, so its coefficient cannot be reduced or set to zero, because we include this variable to account for differences between contests in the probability that an idea gets shortlisted. Predictions for $contest$ are denoted $p_{contest}$ and take values between 0 and 1.
4. To get an *overall* out-of-sample performance metric, concatenate every $p_{contest}$ into a single vector of predicted probabilities $p_{Contests}$. The length of $p_{Contests}\ is$ equal to
$$\sum_{contest \in Contests} nid_{contest}$$
which in words is the grand total of ideas across all 21 contests.
5. To measure the performance of $p_{Contests}$ against the dependent variable out of sample, define $y_{contest}$ as the vector of labels, or the binary vector indicating whether each idea in $contest$ was shortlisted. Then $y_{Contests}$ is analogous to $p_{Contests}$. The length of $y_{Contests}$ is equal to the length of $p_{Contests}$. The number of times the value "1" appears in $y_{Contests}$ is equal to
$$\sum_{contest \in Contests} nshort_{contest}$$
which in words is the total number of shortlisted ideas across all 21 contests.
6. Finally, compute all out-of-sample prediction metrics (ROC, AUC) using $p_{Contests}$ and $y_{Contests}$.

Web Appendix 7: Random Forest

Random Forest (Breiman, 2001) is a well-known method, which uses resampled datasets to reduce prediction variance. We compare it to LASSO here as a benchmark. Random Forest is worse than LASSO. We are unclear why and this issue may be worthy of further research.

Figure WA7.1: (a) ROC Curve for Random Forest and (b) ISE Curve for Random Forest



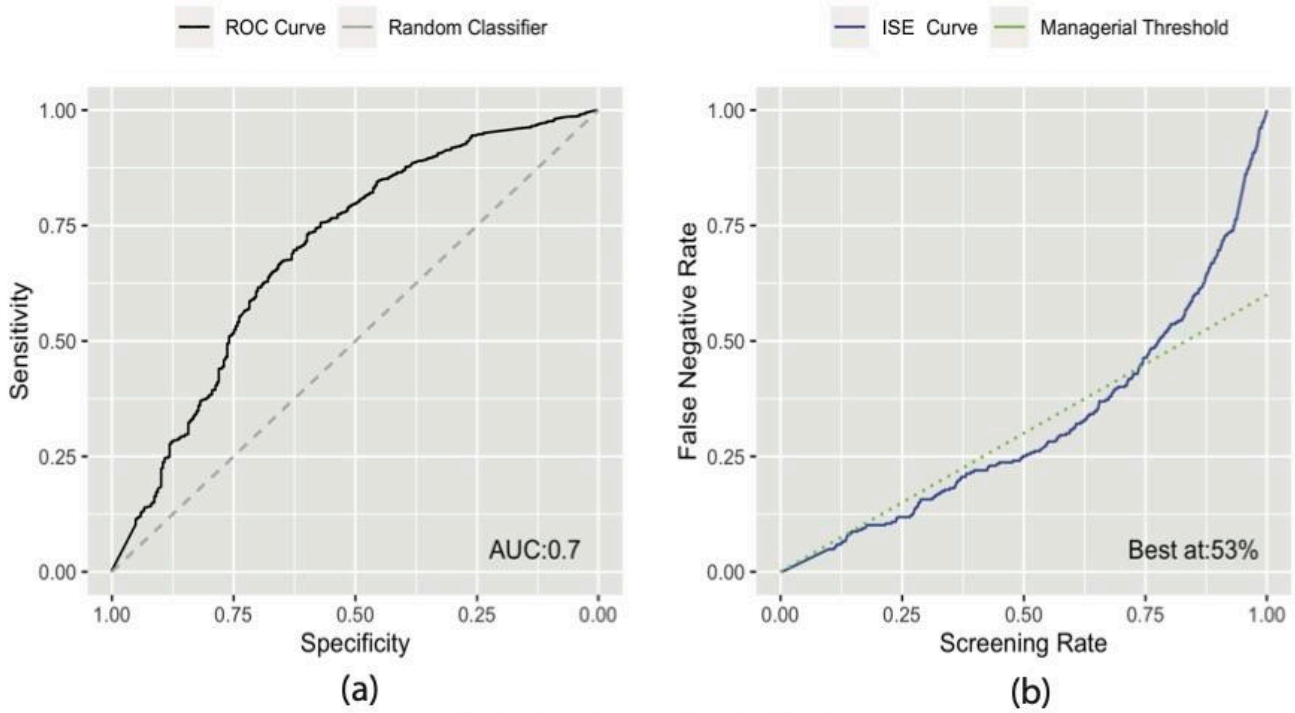
Sensitivity = True Positives / All Positives
Specificity = True Negatives / All Negatives
False Negative Rate = False Negatives / All Positives = 1 - Sensitivity
Screening rate = Number of Ideas Dropped / Number of All Ideas

Web Appendix 8: RuleFit

RuleFit (Friedman and Popescu, 2008) can be thought of as gradient boosted decision trees with LASSO regularization. The approach is meant to sacrifice as little predictive accuracy as possible in return for a set of interpretable decision rules, which come in the form of trees. We will not explore this method in depth here, but only provide results from it for comparison.

Notably, RuleFit performs worse than logistic LASSO. This result may be because the regularization in this model applies to rules instead of variables. This in turn may increase variance and reduce out-of-sample fit in rather noisy datasets, such as ours. While we cannot be certain, this is perhaps the most immediate plausible reason. Further research probing this issue could be useful.

Figure WA8.1: (a) ROC Curve for RuleFit and (b) ISE Curve for RuleFit



Sensitivity = True Positives / All Positives
Specificity = True Negatives / All Negatives
False Negative Rate = False Negatives / All Positives = 1 - Sensitivity
Screening rate = Number of Ideas Dropped / Number of All Ideas

Web Appendix 9: Bayesian Stacking

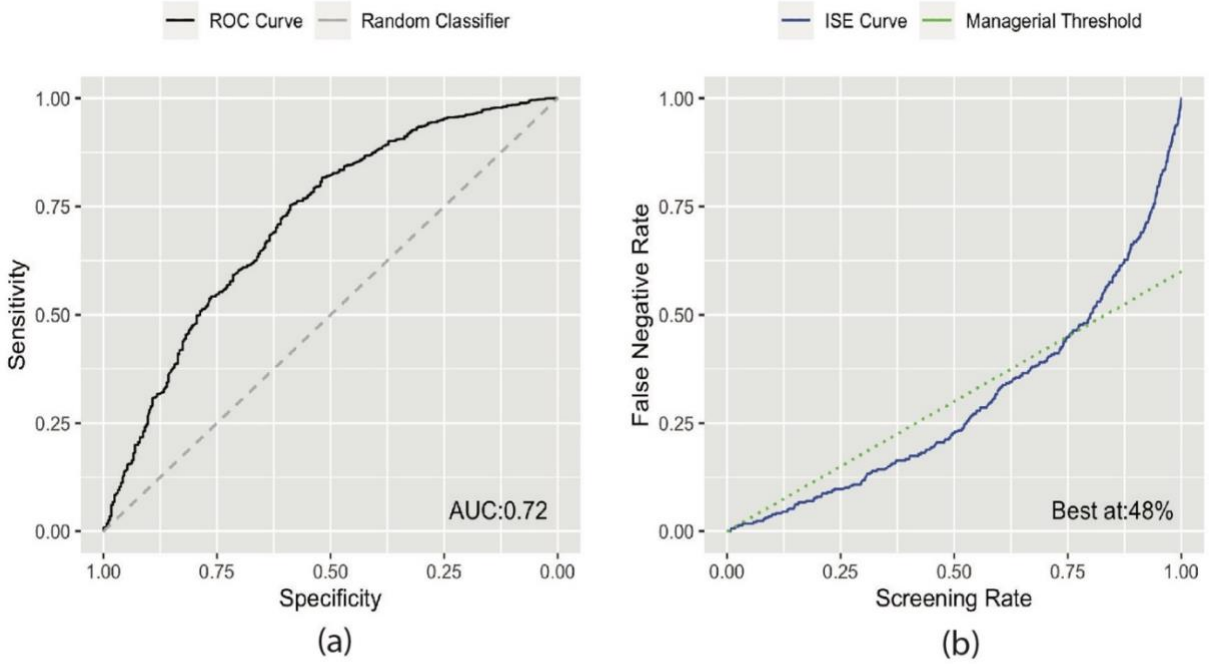
Inspired by the suggestion of an anonymous reviewer of this paper relating to ensemble methods over sets of variables, we applied Bayesian Stacking (Yao et al., 2018). Bayesian Stacking uses weighted averages of the posterior predictive distributions of a group of models. The weights for the weighted average are computed using Leave One Out cross validation in the training sample (The LOO performance is computed using Pareto Smoothed Importance Sampling, PSIS for short. See Yao et al., 2018 and Vehtari et al. 2017 for details.)

If we are to use subsets of variables, we must first decide how to divide the variables into groups, which collection to use as the superset from which to draw, and the number of groups to divide into. There are many ways to split the variables into groups. To cover a wide range of possible stacking configurations, we randomly choose 1, 2, or 3 groups and divide variables from the original set into disjoint subsets. We use each separate group as predictors in separate Bayesian regressions, and then combine the resulting models using Bayesian stacking. By repeating this random configuration and prediction process many times, we can learn how performance varies across different variable groupings. This exploration is not meant to be exhaustive, but rather to understand how Bayesian Stacking typically compares in performance to LASSO on the full set of variables.

As reported above, the best performing configuration achieves an AUC of 0.72, and uses three groups. Word Colocation and Inspiration Redundancy are in the first group. Word Count is alone in the second group. Word Atypicality and Topic Atypicality are together in the third group.

This experiment demonstrates that for most configurations, Bayesian Stacking ensembles of variables are no more accurate than using a simple LASSO regression. A major additional benefit of LASSO is computation speed.

Figure WA9.1: (a) ROC Curve for Bayesian Stacking and (b) ISE Curve for Bayesian Stacking



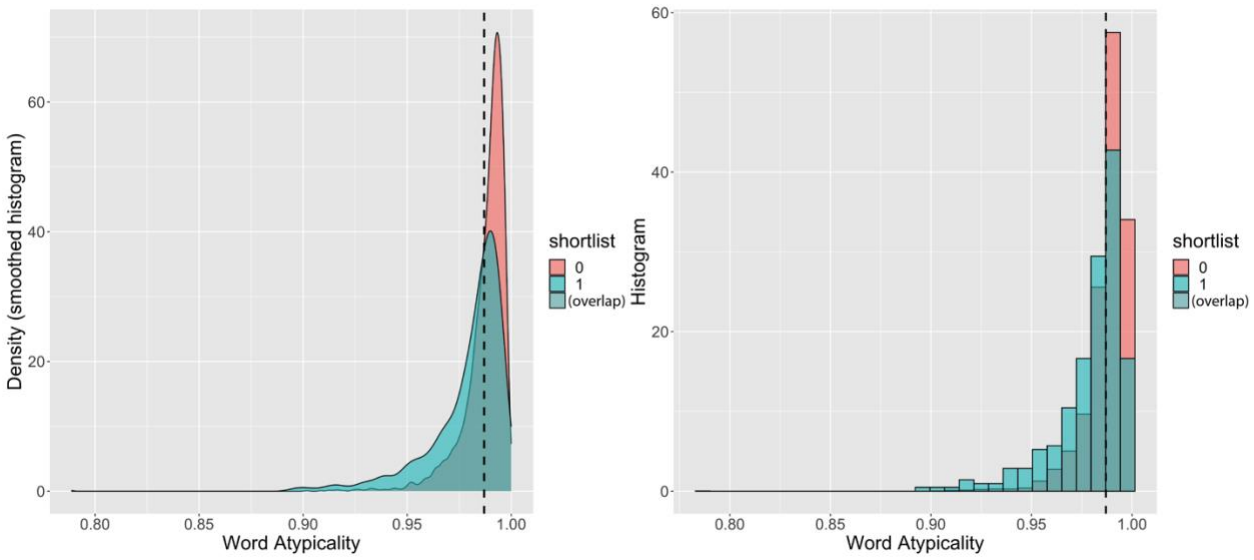
Sensitivity = True Positives / All Positives
Specificity = True Negatives / All Negatives
False Negative Rate = False Negatives / All Positives = 1 - Sensitivity
Screening rate = Number of Ideas Dropped / Number of All Ideas

Web Appendix 10: Descriptive ("Model Free") Evidence About the Performance of Word Atypicality

When all variables are in the model, surprisingly, LASSO selects only Word Atypicality as a predictor, sometimes complimented by another predictor, but in most cases not.

Figure WA10, left and right, presents the empirical distributions of Word Atypicality by Shortlist Status. The x-axis shows the values of Word Atypicality. The y-axis shows a smoothed histogram of density (left) and the empirically observed number of ideas (right). The dotted line represents the overall mean. The green distribution (lower peak) shows the distribution of shortlisted ideas. The red distribution (higher peak) shows the distribution of non-shortlisted ideas. The distribution of Word Atypicality for shortlisted ideas shows that it skews to the left (higher overlap with corpus) compared to the distribution of ideas which were not shortlisted.

Figure WA10. Word Atypicality: Density (left) and Number of Ideas (right) by Shortlist Status



Web Appendix 11: Two-Step Approach

- The procedure for the two-step approach is as follows:

Step 1:

1. Use one contest at a time as a holdout, say c_h
2. For the remaining contests, i.e., the complement (say c_hc) fit these two logistic regressions:
    a. $\log\left(\frac{y_i}{1-y_i}\right) = PS_{c_i} + \beta_1 TA_i + \epsilon_i$
    b. $\log\left(\frac{y_i}{1-y_i}\right) = PS_{c_i} + \alpha_1 WC_i + \nu_i$
    c. Where y_i is shortlist status for idea i, PS_{c_i} is the number of shortlisted ideas in idea i's contest (c_i) divided by its total ideas, TA_i is the Topic Atypicality as measured in Berger & Packard (2018) for idea i, and WC is the Word Count of idea i.
3. Using those two regressions, rank-order the ideas in the holdout contest c_h and create indicators of the bottom 25%, $\mathbb{I}_{TA}(i)$ and $\mathbb{I}_{WC}(i)$. Where $\mathbb{I}_{TA}(i)$ is 1 if idea i is in the bottom 25% according to the model with TA_i and zero otherwise, and $\mathbb{I}_{WC}(i)$ is analogous.
4. Screen out ideas in contest c_h which are in the intersection between bottom 25% ideas from both predictors. Mathematically, screen out the ideas i in the set: $\{i \mid \mathbb{I}_{WC}(i) = 1 \text{ and } \mathbb{I}_{TA}(i) = 1\}$.

Step 2:

With the reduced dataset, compute the Percent Screened Before Losing the First Winner as in the original procedure. Predictors are Word Colocation and the Percent of Ideas Counted as First Place Winners (in some contests multiple ideas tied for the winner). We screen out the first winner by creating a matrix where each row corresponds to an idea and has two fields: the out-of-sample prediction probability of shortlisting for that idea and the observed outcome (not seen by the model which made the prediction). This matrix is rank ordered in ascending order of predicted probability of being shortlisted. The first winner screened out is the index of the first winner in that ordering.

The first contest-by-contest step screens out 317 ideas. The first winner rank is 547 after removing the ideas from the above first round. The total is 864, or 864 / 4191 ≈ 21%. When using simple word count to screen out ideas in the first step, 864 ideas cannot be screened out without also eliminating two first-place winners.

Web Appendix 12: Survey of Senior Managers of Innovation About the Performance of Our Models

New approaches are most useful if the results encourage firms to use them (Little 2004). To show usefulness, we survey managers' views on AI-based approaches and their willingness to adopt our models. To do so, we conduct eight semi-structured interviews with senior innovation managers from large companies like Airbus, BMW, Continental, Siemens, Canon, or Voest. We only select managers for interviews who deal with idea evaluation and selection from internal or external crowdsourcing contests or other ideation challenges. The managers have on average around 15 years of experience in innovation management.

We present managers with our key results: the two-step approach, which can screen out as much as 21% of estimated bad ideas without sacrificing a single winner The model that contains all predictors can screen out 44% of all ideas while sacrificing only 14% of the shortlisted ideas. The interviews last between 20-35 minutes. The interviews were recorded but are confidential. Below are the main findings.

Relevance and Current State of Idea Screening. All eight managers consider idea evaluation and selection an important task. Yet, they all point out that idea selection involves twin problems of tremendous effort and personal biases. To screen ideas to reduce these problems, some managers already turn to the use of naïve screening heuristics like number of comments or likes.

AI in Idea Screening. So far, none of the eight managers applies AI in idea screening. Yet, all managers express strong interest in AI models to improve idea evaluation and selection. They are also strongly interested in testing our models because of their suitability, adaptability, and better performance than their current manual procedures.

Performance of Our Models. All the managers see clear value in the overall performance of our two-step approach: screen 21% of estimated bad ideas without sacrificing a winner. However, the majority prefer the best performance of our single model approach: screen out 44% of estimated bad ideas for a sacrifice of only 14% of shortlisted ideas. This expressed preference suggests that managers' loss function for sacrificing winners is not very steep. Managers explain that they lack resources to follow-up

on most good ideas. While identifying a creative idea is very important, in the final analysis, other factors are also critical to transforming a selected idea into business value. These factors are team composition, plan implementation, management support, and organizational commitment.

Other Benefits. Managers mention other benefits that AI could provide for ideation: rank ordering ideas, or screening out duplicate or similar ideas, and estimating the probability of an idea being a winner. With a few adjustments, our models can provide all of these.

Summary. The idea selection process is crucial in the new product development process, but managers point out that idea selection involves twin problems of tremendous effort and personal biases. They are strongly interested in testing our models because of their suitability, adaptability, and strong performance. Their loss function for sacrificing winners is not steep. While the idea itself is very important, other factors are also critical to transforming a selected idea into business value.

Web Appendix 13: LASSO Code

The following R code runs the LASSO cross-validation procedure described in the methods section.

```
source('lasso_fun.R')
main_data_path <- "master_data.csv"
vars <- c("peer_dev_jacc")
data_object <- data_make(input_file=main_data_path,
    varnames =vars, filter_on_X='impute', use_winner = F)

X <- data_object$X
orig_X_names = colnames(X)
X_orig <- data_object$X_orig
idea_df <- data_object$idea_df
contest <- idea_df$contest
shortlist <- idea_df$shortlist
winner <- idea_df$winner
pen_vec <- c(rep(1, ncol(X) - 1), 0)

lasso_is <- get_model(X, X, contest, shortlist, T, T, pen_vec)
oos_yhat <- rep(NA, length=0)
oos_y_test <- rep(NA, length=0)
oos_winner = rep(NA, length=0)
oos_idea_id = rep("", length=0)
for (i in 1:length(levels(contest))) {
    contest_i <- levels(contest)[i]
    train_ind <- contest != contest_i
    test_ind <- contest == contest_i

    setwd(paste0('./', contest_i))
    lasso <- get_model(X[train_ind, ], X[test_ind, ],
    contest[train_ind], shortlist[train_ind],
          display=T, save_coefs=F, pen_vec=pen_vec,
          winner[train_ind])
    oos_yhat <- c(oos_yhat, lasso$yhat)
    oos_winner = c(oos_winner, winner[test_ind])
    oos_idea_id = c(oos_idea_id,
     as.character(idea_df$idea_id[test_ind]))
    oos_y_test <- c(oos_y_test, shortlist[test_ind])
    setwd('..')
}

setwd('../Output')
results <- list(observed=oos_y_test, yhat=oos_yhat, X=X,
X_orig=X_orig, idea_df=idea_df)
res_path = paste0("lasso_res_", vars[1], "_", length(vars),
".RData")
```

```
save(results, file=res_path)

library(glmnet)
library(pROC)
library(foreach)
library(DMwR2)
library(MLmetrics)
library(doParallel)
library(yardstick)
numCores <- detectCores()
registerDoParallel(numCores)

data_make <- function(input_file, varnames=c("*nf*", "*jm*",
"ks*", "tc", "cc", "cc_orig", "degree", "peer_dev*", "burt"),
    filter_on_X='impute', use_winner=F) {
    idea_df <- read.csv(input_file, header=T,stringsAsFactors=T)
    contest <- idea_df[, names(idea_df)=="contest"]
    col_list <- list()
    for (i in 1:length(varnames)) {
        var_i <- varnames[i]
        grep_target <- glob2rx(var_i)
        col_list[[i]] <- grep(grep_target, names(idea_df))
    }
    X <- as.matrix(idea_df[, unlist(col_list)])
    if (filter_on_X == 'impute') {
        getmode <- function(v) {
            uniqv <- unique(v)
            uniqv[which.max(tabulate(match(v, uniqv)))]
        }
        if ('cc' %in% colnames(X)) {
            cc_idx <- which(colnames(X) == 'cc')
            X_cc <- X[, cc_idx]
            X_cc[is.na(X_cc)] <- getmode(X_cc)
            X[, cc_idx] <- X_cc
        }
        if ('cc_orig' %in% colnames(X)) {
            cc_idx <- which(colnames(X) == 'cc_orig')
            X_cc <- X[, cc_idx]
            X_cc[is.na(X_cc)] <- getmode(X_cc)
            X[, cc_idx] <- X_cc
        }
        if (dim(X)[2] == 1) {
            non_cc <- 1
        } else {
            non_cc <- which(!(colnames(X) %in% c('cc',
               'cc_orig')))
        }
```

```
        for (k in 1:length(non_cc)) {
            spot_k <- non_cc[k]
            X_spot_k <- X[, spot_k]
            X_spot_k[is.na(X_spot_k)] <- median(X_spot_k,
               na.rm=T)
            X[, spot_k] <- X_spot_k
        }
    } else if (filter_on_X == 'drop') {
        X_filter <- rowSums(is.na(X)) == 0
        idea_df <- idea_df[X_filter, ]
        X <- X[X_filter, ]
    }
    if (ncol(X) > 1) {
        nan_cols <- colSums(is.nan(X)) > 0.1 * nrow(X)
        X <- X[, !nan_cols]
    } else if (ncol(X) == 1) {
        colnames(X) <- varnames
    }
    X_orig <- X
    X <- scale(X)
    X <- add_contest_short_prob(X, idea_df, use_winner =
use_winner)
    return(list(idea_df=idea_df, X=X, X_orig=X_orig))
}


add_contest_short_prob <- function(X, idea_df, use_winner=F) {
    contest_names <- levels(idea_df$contest)
    prob_vec <- rep(0, nrow(X))
    for (i in 1:length(contest_names)) {
        contest_i <- contest_names[i]
        contest_idx <- idea_df$contest == contest_i
        prob_vec[contest_idx] <-
               sum(idea_df$shortlist[contest_idx]) /
               sum(contest_idx)
        if (use_winner) {
           prob_vec[contest_idx] =
               sum(idea_df$winner[contest_idx] ==1) /
               sum(contest_idx)
        }
    }
    colnames0 <- colnames(X)
    X <- cbind(X, prob_vec)
    colnames(X) <- c(colnames0, "short_prob")
    return(X)
}
```

```
get_model <- function(X, newx, contest_vec, y_train,
        display=F, save_coefs=F, pen_vec=rep(1, ncol(X)),
               winner) {
    X_drop <- ifelse(is.na(rowSums(X)), T, F)
    X <- X[!X_drop, ]
    contest_vec <- contest_vec[!X_drop]
    y_train <- y_train[!X_drop]
    NC <- length(unique(contest_vec))
    contests <- unique(contest_vec)

    lasso <- glmnet(X, as.factor(y_train), alpha=1,
               family="binomial", penalty.factor = pen_vec)
    grid <- lasso$lambda

    cv_obj <- foreach(l=1:length(grid), .combine=c) %dopar% {
        pmetric_l <- rep(NA, NC)
        for (i in 1:NC) {
            holdout <- contest_vec == contests[i]
            X_i <- X[!holdout, ]
            X_holdout <- X[holdout, ]
            y_holdout <- y_train[holdout]
            y_i <- y_train[!holdout]
            fraction_0 = rep(1 - sum(y_i) / length(y_i), sum(y_i
               == 0))
            fraction_1 <- rep(1 - sum(y_i == 1) / length(y_i),
               sum(y_i == 1))
            glm_wts = numeric(length(y_i))
            glm_wts[y_i == 0] = fraction_0
            glm_wts[y_i == 1] = fraction_1
            model_i <- glmnet(X_i, as.factor(y_i), alpha=1,
                family="binomial",
                penalty.factor = pen_vec, weights=glm_wts)
            yhat_li <- predict(model_i, newx=X_holdout,
                s=grid[l], type="response")
            coefs_li <- predict(model_i, s=grid[l], type="coef")
            nvars_li <- sum(coefs_li != 0)
            #roc_obj <- roc(y_holdout, c(yhat_li), quiet=T)
            #pmetric_li <-roc_obj$auc
            y_holdout_temp <- y_holdout[order(c(yhat_li))]
            pmetric_li <- sum(which(y_holdout_temp == 1))
            #y_holdout_temp <- winner[order(c(yhat_li))]
            #pmetric_li = which(y_holdout_temp == 1)[1]
            y_bin_pred <- ifelse(yhat_li > quantile(yhat_li,
               0.25), 1, 0)
            y_bin_pred <- factor(y_bin_pred, levels=c("0", "1"))
            pmetric_l[i] <- ifelse(nvars_li > 2, pmetric_li, 1)
```

```
        }
        mean_pmetric_l <- mean(pmetric_l)
        mean_pmetric_l
    }

    pmetric <- cv_obj
    dg_flag <- rep(F, length(grid))
    for (i in 1:length(grid)) {
        coefs_temp <- predict(lasso, s=grid[i], type="coef")
        nvars_temp <- sum(coefs_temp != 0)
        dg_flag[i] <- ifelse(nvars_temp > 2, F, T)
    }
    grid_ndg <- grid[!dg_flag]
    pmetric_ndg <- pmetric[!dg_flag]
    best.lambda <- grid_ndg[which.max(pmetric_ndg)]
    if (display) print(predict(lasso, s=best.lambda,
          type="coef"))
    if (save_coefs) {
        a <- predict(lasso, s=best.lambda, type="coef")
        a <- as.data.frame(round(as.matrix(a), 2))
        rownames(a) <- change_names(rownames(a))
        a$vars <- rownames(a)
        names(a)[1] <- "coefs"
        a <- a[a$coefs != 0, ]
        a <- a[order(abs(a$coefs), decreasing=T), ]
        write.csv(a, "lasso_coefficients.csv")
    }
    yhat <- predict(lasso, newx=newx, s=best.lambda,
          type="response")
    return(list(yhat=yhat, lasso=lasso,best.lambda=best.lambda))
}
```

Web Appendix 14: Two-Stage Model Code

The following R code runs the two-stage procedure described in the methods section.

```
source('lasso_fun.R')
main_data_path <- "master_data.csv"
vars <- c("wo_goog", "wc", "peer_dev_lda", "peer_dev_jacc",
     "ks", "min_nf")
data_object <- data_make(input_file=main_data_path,
               varnames =vars, filter_on_X='impute',
               use_winner=T)

X <- data_object$X
Xso <- X
X = X[, c(5, 7)]
X_orig <- data_object$X_object
idea_df <- data_object$idea_df
contest <- idea_df$contest
shortlist <- idea_df$shortlist
winner <- idea_df$winner
pen_vec <- c(rep(1, ncol(X) - 1), 0)

oos_yhat <- rep(NA, length=0)
oos_y_test <- rep(NA, length=0)
oos_doomed <- rep(NA, length=0)
oos_winner = rep(NA, length=0)
total_dropped = 0
winners_dropped = 0
for (i in 1:length(levels(contest))) {
    contest_i <- levels(contest)[i]
    train_ind <- contest != contest_i
    test_ind <- contest == contest_i
    setwd(paste0(contest_i))
    df_so_i <- data.frame(y=shortlist[train_ind],
          Xso[train_ind,])
    so_i1 <- glm(y ~ peer_dev_lda, data=df_so_i,
          family="binomial")
    so_i2 <- glm(y ~ wc, data=df_so_i, family="binomial")
    quantile_i = 0.25
    X_test_so_i <- data.frame(Xso[test_ind, ])
    yhat_so_i1 <- predict(so_i1, X_test_so_i, type="response")
    yhat_so_i1 <- ifelse(yhat_so_i1 < quantile(yhat_so_i1,
          quantile_i), 1, 0)
    yhat_so_i2 <- predict(so_i2, X_test_so_i, type="response")
```

```
    yhat_so_i2 <- ifelse(yhat_so_i2 < quantile(yhat_so_i2,
          quantile_i), 1, 0)

    yhat_so_i <-  yhat_so_i1 * yhat_so_i2
    total_dropped <- sum(yhat_so_i) + total_dropped
    df_contest_i <- idea_df[test_ind, ]
    target_bin_i <- ifelse(df_contest_i$winner[yhat_so_i == 1]
          == 1, 1, 0)
    if (sum(target_bin_i) > 0) {
        cat("WINNER DROPPED FOR ", contest_i, "\n")
        winners_dropped = winners_dropped + sum(target_bin_i)
    }
    lasso <- get_model(X[train_ind, ], X[test_ind,],
        contest[train_ind], shortlist[train_ind],
        display=T, save_coefs=F, pen_vec=pen_vec,
          winner[train_ind])
     oos_doomed <- c(oos_doomed, yhat_so_i)
     oos_yhat <- c(oos_yhat, lasso$yhat)
     oos_y_test <- c(oos_y_test, shortlist[test_ind])
     oos_winner = c(oos_winner, winner[test_ind])
    setwd('..')
    cat('\n\n')
}
winner2 <- oos_winner[oos_doomed==0]
yhat2 <- oos_yhat[oos_doomed==0]
widx2 = which(winner2[order(yhat2)] == 1)
widx = which(oos_winner[order(oos_yhat)] == 1)
garbage_dropped_n = total_dropped + widx2[1]
pct_garbage_dropped = round(garbage_dropped_n / nrow(X), 2) *
100
cat("Can screen out ", garbage_dropped_n, " ideas w/o losing
winners.
This is ", pct_garbage_dropped, "% of ideas.")
   oos_yhat = ifelse(oos_doomed == 1, 0, oos_yhat)

oos_yhat = ifelse(oos_doomed == 1, 0, oos_yhat)
cat("AUC: ", roc(oos_y_test, oos_yhat)$auc)
setwd('../Output')
results <- list(observed=oos_y_test, yhat=oos_yhat, X=X,
     X_orig=X_orig, idea_df=idea_df)
res_path = paste0("../Output/twostage_res_", vars[1], "_",
length(vars), ".RData")
save(results, file=res_path)
```

Web Appendix 15: ISE Curve Code

The following R code produces an ISE curve using the outputs of the script in Web Appendix 13.

```
ise_plot <- function(roc, observed, yhat, savepath="ise.png") {
    fnrs <- 1 - roc$sensitivities
    n_dropped <- rep(NA, length(fnrs))
    for (i in 1:length(n_dropped)) {
        n_dropped[i] <- sum(yhat < roc$thresholds[i], na.rm=T)
    }
    pct_drop <- n_dropped / length(yhat[!is.na(yhat)])
    fnr_df <- data.frame(FNR=fnrs, Pct_dropped=pct_drop)
    best_idx <- which.min(fnrs - pct_drop * 0.6)
    best_pct <- round(pct_drop[best_idx], 2) * 100
    g2 <- ggplot(data=fnr_df, aes(x=Pct_dropped, y=FNR)) +
        geom_line(aes(color="ISE Curve")) +
    geom_segment(aes(x=0, xend=1, y=0,
        yend=0.6, color="Managerial Threshold"),
        linetype="dotted") + xlab("Screening Rate") +
        ylab("False Negative Rate")
        g2 <- g2 + annotate("text", x=0.85, y=0.05,
            label= paste0("Best at:", best_pct, "%"))
        g2 <- g2 + scale_colour_manual("",
            breaks = c("ISE Curve", "Managerial Threshold"),
            values = c("blue", "green")) +
            theme(legend.position = "top")
        ggsave(savepath, g2, width=4, height=4, units="in")
        sacrifice = round(fnrs[best_idx], 2) * 100
        return(list(best_pct=best_pct, best_idx=best_idx,
            sacrifice=sacrifice))
    }
ise_object = ise_plot(roc, observed, yhat)
```

References Web Appendix

Little, JDC. 2004. "Models and Managers: The Concept of a Decision Calculus." *Management Science.* 50(12): 1841-1853.Sun, L, Zheng, X, Jin, Y, Jiang, M, and Wang, H. 2019. "Estimating Promotion Effects Using Big Data: A Partially Profiled LASSO Model with Endogeneity Correction." *Decision Sciences. 50*(4): 816-846.

Tanczos, E, Nowak, R, and Mankoff, B. 2017. "A KL-LUCB Bandit Algorithm for Large-Scale Crowdsourcing." *Proceedings of the 31st International Conference on Neural Information Processing Systems.* December: 5896-5905.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and WAIC. *Statistics and computing*, *27*(5), 1413-1432.

Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Using Stacking to Average Bayesian Predictive Distributions (With Discussion). Bayesian Analysis, 13(3), 917-1007.