Kaleidoscope: Computer-Assisted Ideation through Idea Network Exploration

Zifei Shan, Yuke Zhu and Tianxin Zhao*
Department of Computer Science, Stanford University
{zifei,yukez,tianxin}@stanford.edu

ABSTRACT

Idea generation (ideation) is an essential process in human creativity that leads to great innovation. Brainstorming is one of the most widely-used and powerful ideation methods. In this paper, we propose a novel approach to assisting human brainstorming, which allows computers to collaborate with humans and boost new ideas. To achieve this goal, we first model the brainstorming process in a network structure, then construct *idea networks* using heterogeneous datasets varying in both domains and forms. We define objectives to discover insightful ideas from idea networks, and design algorithms that satisfies these objectives. As an outcome, we develop an online brainstorming system with automated idea recommendation and evaluate our approach by user surveys. To the best of our knowledge, our work is the first attempt to enable human-computer collaboration in an interactive ideation process. Our contribution lies in the following: (1) we initiate the methodology of computer-assisted ideation; (2) we craft objectives and algorithms that shed insight on recommending knowledge in information networks; (3) we build a system to facilitate ideation with knowledge and evaluate its effectiveness.

Keywords

Information Networks, Data Mining, Model, Algorithm, Collaborative Innovation, Ideation, System

1. INTRODUCTION

Ideation is an essential process in creativity and problem-solving. In a typical collaborative ideation scenario, people spend hours or days brainstorming ideas. The quality of brainstorming largely determines the outcomes. In most cases, ideation involves human beings exclusively, without assistance from other techniques. The field of computer-assited ideation has an untapped potential to be explored. Our work builds upon the following observations of the ideation process:

1. **Ideas build on each other.** Ideas interact with each other. Computers suggest ideas related to people's original ideas, and people can further develop these ideas to better ones.
2. **Deferring judgment**. In the ideation phase, as long as related to the topic, ideas should not be judged. Judgments should be made in subsequent phases. This encourages people to suggest "wild" ideas without considering whether it is good or not.
3. **Seeking quantity and diversity.** During an ideation process, a large quantity of ideas should be encouraged, and ideas from different disciplines better contribute to a novel solution.

*Group 12, CS 224W 2013 course project final report.

Based on these observations, we propose a methodology to apply network science to help ideation. First, we define the *idea network* model, including *Problem Idea Network (PIN)* that captures all the ideas in one specific human brainstorm process, and *Backend Idea Networks (BIN)* that include all ideas in the knowledge bases our system hold. As the PIN evolves, we want to suggest ideas from BINs that users may be interested in. To ensure quality of suggestions, we define objective functions to capture the score of potential suggestions, given a snapshot of the PIN. The objective balances over how well the suggestions cover diverse topics, and how well they combine existing ideas.

We propose our system, KALEIDOSCOPE, as an integrated solution for a computer-assisted ideation. To craft the system and recommend ideas based on the objective, we first construct idea networks on various datasets based on existing relationships and similarity measures. Second we detect communities in BINs, to serve as a foundation for evaluating the diversity of ideas. Then we study how to map each node in PINs to a most similar node in BINs, using general information retrieval approaches. Finally, we analyzed the optimum algorithm and two greedy algorithms, and implement a greedy approach, which can effectively achieve a heuristic in our objective function.

We further conduct a survey-based evaluation, aggregating ideas of module tests and blind tests, to evaluate our overall system and objective functions, and give us further insights about the ideation process.

Our contribution lies in following aspects:

1. We initiate the methodology to help people "breakout" and "unstuck" in ideation by interacting with computers.
2. Our objectives and algorithms can be generalized to retrieve useful knowledge in information networks.
3. We build a working system to help ideation with heterogeneous knowledge sets, qualitatively evaluate its effectiveness, and generate valuable insights.

2. RELATED WORK

Due to the rapid growth of the Internet as well as recent development in computing and storage technologies, massive amount of data, which we usually refer to "Big data", has been collected and processed in a unprecedentedly higher degree of complexity. Researchers estimated that humankind was able to store 2.9×10^{20} bytes of data, communicate nearly 2×10^{21} bytes during 2007; meanwhile, the annual growth rate of general-purpose computing capacity stays 58% [12]. The augmenting data indicates an incredible potential of knowledge discovery and information retrieval, which can be utilized to facilitate the human ideation process.

2.1 Information Networks

In recent years, many attempts have been made to harvest, integrate and analyze general, structured data in a network model. Online encyclopedias (e.g. Wikipedia) aim at structuring general Web information [23]. The Semantic Web is a well-known example that seeks structured, semantic understanding on top of Web pages [4]. While, these sources often lack a well-organized structure to retrieve the most relevant information snippets from long, full-text documents. Some efforts have been made to structure general human knowledges into a uniform, evoling database [16, 5, 7]. However, much of the work relies on tremendous human involvement to establish database entities and relations. In addition, knowledge collected in these databases is limited to a small set of topics (e.g. about 68.7% of total 39 million topics in Freebase are about *Music* [10]).

Another drawback of knowledge bases is that the majority of knowledge is far from sophisticated. A typical example of facts NELL recently learned can be "*atmospheric_wind* is a *weather phenomenon*", or "*magnificent_birds* is a *bird*" [18] [1]. We found most of these facts can hardly bring any insights into human ideation.

2.2 Web Extraction Systems

Web extraction systems focus on eliminating the human involvement and automating the analysis and extraction process of large data collections. A few successful examples can be seen in [26, 3, 9], which possess the power to extract named-entities and entity relations without human supervision. Besides the limited set of relations these extraction systems capture, they often lack a graph-based structure, which is essential for retrieving supplementary relevant information. Similar to information networks, web extraction systems interact with end-users in a passive manner. They, like other query-based retrieval systems (Google, Bing, etc.), fetch results based on user inputs, but cannot interactively recommend high-quality information to users. Consequently, neither of them performs well in the context of computer-assisted ideation.

2.3 Information Maps

The work closest to ours is information maps [19] which automatically establish relations among retrieved information snippets and allow users to alter the maps to better reflect their interests. The idea of information maps has been successfully applied in the domains of news articles and research papers [21, 20]. Similar techniques can be adopted to organize information into an idea network structure. However, information maps only focus on a unique type of data, either news article or research papers, but not both. While in ideation process, heterogeneous data from various sources is preferable in order to throw light on the existing ideas. To the best of our knowledge, this is the first time that a network integrating various information sources (idea network) is used to assist human ideation process.

In the work of information maps, the authors proposed three objectives for creating a good map [19], which inspire our work: *Coherence* contributes to a smooth storyline, *Coverage* contributes to the overall diversity of the story, and *Connectivity* gives insight of the connections of different stories.

2.4 Computer Aided Innovation

Computer Aided Innovation (CAI) has been covered broadly in marketing and product management. CAI involves all aspects in an innovation. Categories of CAI includes idea management, patent management, strategy management and etc. In the field of ideation, they emphasize on managing ideas rather than suggesting new ideas. Although CAI software has been put into practice since 1990, no work has been done in order to suggest useful ideas to users, especially with a novel prospective to capture ideas and their connections in a network model. The most related software called *IdeaFisher* [6] was available online since 1980s. Their data comes from two datasets storing phrases and questions respectively. However, instead of suggesting new ideas, *IdeaFisher* simply presents all its semantically related concepts as well as retrieves queried images from search engines, which makes the software hardly perform well in real-world ideation situations.

[1]These facts are learned by NELL in September 27, 2013

3. DATASET SELECTION

Since quantity and diversity are essential in ideation, we select large, heterogeneous datasets in different forms to construct a diversified knowledge base for idea recommendation. Our datasets include: basic concepts, images and encyclopedias.

ConceptNet. For basic concepts, we choose ConceptNet [15], which is a semantic network with basic knowledge of words and phases. Intuitively, when users get stuck in some ideas, it can be helpful to simply suggest related semantic meanings. In addition, the network is also useful for combining multiple ideas and finding semantically related concepts to help people get unstuck.

DBpedia. For encyclopedias, we use Wikipedia [23], specifically a structured English language dataset DBpedia [2] extracted from Wikipedia. The articles in DBPedia offers exhaustive descriptions of knowledge and concepts with rich references to other related articles.

MIRFLICKR. For images, we use a labeled dataset from Flickr provided by MIRFLICKR [13], which consists of 25,000 labeled images. Annotations of images help us understand and retrieve images in the network. Images convey insightful information to users and stimulate further ideation. The adoption of an image idea network demonstrates that our proposed objectives and algorithms in finding a good suggestion is generally applicable in different heterogeneous networks.

We build networks and detect communities in these datasets, for our idea recommendation. However, as these datasets are very large (especially for community detection), we subsample the DBPedia network, and apply heuristics to ConceptNet, to reduce the number of stored edges to a managable scale, which is described in detail in Section {sec:eval}.

4. MODEL

4.1 Definition: Idea Networks

First we formally define idea networks, and justify the naming by its special attributes.

- A **node** in an idea network is an **idea**, which is defined as *a piece of information snippet that gives insight or conveys thoughts in an ideation process.*
- An **edge** in an idea network is *a connection between two ideas.*

We pick the term "idea networks" to emphasize our use case in ideation process. The definition of ideas is relatively vague, and one might argue that idea network is identical to general information networks. However, our idea networks differ from information networks due to the special property of the ideation process: ideas are *vague* in nature. An idea might or might not contain useful information, or might convey ambiguous information based on different interpretations. Being vague helps us to embrace quantity and breakouts, and enable further ideas to build on top of previous ones.

Armed with the definition of idea network, we further define computer-assisted ideation process:

- The **Problem Idea Network (PIN)** is the idea network of the current problem that users try to solve. It starts with a *problem*, and grows as ideas are added into the network. The goal of brainstorming is to enrich this network with many diverse ideas and eventually find useful ideas for problem solving.

- **Backend Idea Networks (BINs)** are idea networks in our system, extracted from online datasets. These networks function as a knowledge base where our system retrieves ideas to recommend.

- In our **computer-assisted ideation**, given a snapshot of PIN at some timepoint, our system maps each idea in PIN to a most similar node in BIN, and get the so-called *Mapped PIN* as a subgraph of BIN. Then the system use Mapped PIN's information to retrieve nodes in BINs for recommendation. To simplify naming, *PIN* mentioned in this section below refer to *Mapped PIN*.

4.2 Objectives for Idea Suggestion

Now that we have mapped users' ideas (Mapped PIN) in a Backend Idea Network, how to find ideas and add them to Problem Idea Network? Which ideas might be the most desired by users?

Our goal is to recommend a set of nodes in BIN that maximizes some objective functions related to Mapped PIN. To craft out these objectives, we start with some intuitions of brainstorming processes. We thus propose "3C" objectives: Connections, Coverage and Combination. In the evaluation part, we tune parameters to balance these three objectives.

4.2.1 Connection

Based on the observation that *ideas are built on each other,* we want to recommend ideas that are related to existing ones in PIN, rather than arbitrary ideas. A related idea helps keep the brainstorming on track and brings further thoughts to the ongoing process.

In order to enable recommended ideas "go into depth", one suggestion can build on another suggestion, and they are "connected" to PIN as long as there is a *path* that connects these suggestions (denoted as SUGG) to PIN, or in other words, the network $PIN \cup SUGG$ is a connected component.

We use connection as a constraint, and optimize other objectives while ensuring connections. The connection constraints can be described as below, where *NumCC* denotes the number of connected components in a network:

$$NumCC(SUGG \cup PIN) = 1 \tag{1}$$

4.2.2 Coverage

Based on the observation that *diversity is encouraged*, we recommend ideas that let users "jump out of the loop" and break out into a new area. Better ideas cover topics that users haven't thought about before.

To quantify this aspect, we first detect communities in BINs, and measure how well the current PIN with suggested ideas covers these communities in BINs. We maximize the *incremental coverage* when suggesting a set of new ideas. Intuitively, we want to suggest ideas in BIN-communities that have not or seldom been covered in PIN; If a community is already well-covered by many ideas in PIN, we prefer ideas in some other communities. Our objective function for coverage is inspired by [22]:

$$cover_{PIN}(w) = 1 - \prod_{idea \in PIN} (1 - cover_{idea}(w)) \tag{2}$$

$$Cover(PIN) = \sum_{w} \lambda_w \, cover_{PIN}(w) \tag{3}$$

In above definitions, w is a community and $idea$ is an idea in PIN. $cover_{idea}(w) \in [0,1]$ is how well an idea covers a community, which can be set to a constant, or set by TF-IDF, or learned from further learning about node and community content.

$cover_{PIN}(w)$ denotes how well PIN covers a single community w, and the total coverage function $Cover(PIN)$ is a weighted sum of coverage of different communities. Weights of communities, λ_w, is **Community Relevance** of the problem, which can be dynamically learned and adjusted in brainstorming.

4.2.3 Combination

In brainstorming, one golden rule is seeking combination [14]. It always gives a new way of thinking when combining two ideas together. To measure whether an idea is a *combination* of ideas in PINs, we simply count the number of links from the idea to all nodes in PIN. However, we do not consider combination as a linear function: The difference between combining *one or two* ideas, is actually more important than that between combining *10 or 11* ideas. Therefore we use a log function to measure combination:

$$comb(idea, PIN) = log \sum_{j \in PIN} q_j \, IsEdge(idea, j) \tag{4}$$

$$Comb(SUGG, PIN) = \sum_{i \in SUGG} log \sum_{j \in PIN} q_j \, IsEdge(i, j) \tag{5}$$

where $IsEdge(i, j)$ is 0 when there is no edge between two ideas, and 1 otherwise. q_j is the **Idea Quality** for node j, based on the assumption that ideas have different quality (or relevance to the problem), and we want to combine useful ideas to generate new ones. This parameter can also be tuned from user feedback for ideas. $comb(idea, PIN)$ is the combination of a single idea, and $Comb(SUGG, PIN)$ is its sum among all nodes in suggestion set.

4.2.4 Adaptive Learning to Adjust Objectives

Some ideas might be not related to the topic, some might be ones that users do not want to think of, and some might be more interesting to users than others. Moreover, objectives might differ by users, domains and problems. To enable more powerful automatic suggestions, users shall be able to specify *Idea Quality* for an idea, and *Community Relevance* in a specific problem for a community. In future work we plan to study how to learn adaptively from users' behaviors and feedback, to dynamically adjust the objectives for better performance.

4.3 Formalizing Objective Function

Having articulated all the objectives, now we combine them to make a balance. As mentioned, connection is used as constraints, while coverage and combination are our scoring functions to maximize. We formulate the optimization problem as below:

PROBLEM 1. *Given all nodes in a Problem Idea Network (PIN) mapped into a Backend Idea Network (BIN): $P = \{v_1, v_2, \ldots, v_n\}$, find K suggested nodes in BIN, $S = \{s_1, s_2, \ldots, s_K\}, s_i \in BIN \, for all \, i \in [1, 2, \ldots, K]$, which maximize the following objective function:*

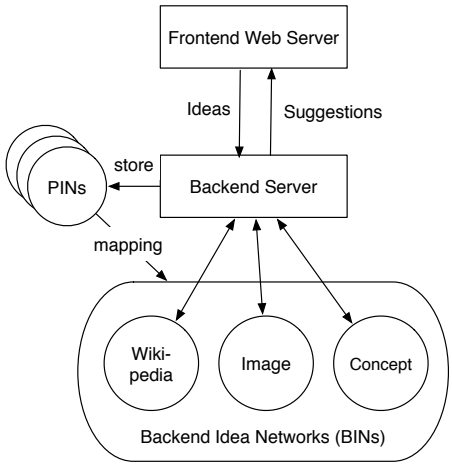


Figure 1: Kaleidoscope System Architecture

$$\max_{W} \ \alpha \ Cover(S \cup P) + \beta \ Comb(S,P) \tag{6}$$

$$s.t. \ NumCC(S \cup P) = 1 \tag{7}$$

We optimize this function with a greedy forward search algorithm in Section 5.

5. ALGORITHMS

In this section we propose our system KALEIDOSCOPE, an integrated solution for idea suggestion. The architecture of our system is demonstrated in Figure 1. The backend server integrates multiple BINs, stores PINs for different problems, accepts ideas from frontend, and gives suggestions based on our recommendation algorithm.

Although the design of frontend and the communication between frontend and backend are relatively simple, the backend components involve careful design and research. In following subsections, we will discuss our approaches in constructing BINs, community detection in BINs, mapping PINs to BINs, and finally a greedy algorithm to optimize the objective function.

5.1 Construct Backend Idea Networks

We build multiple idea networks for each dataset, rather than aggregating them into a uniform, large network. This simplifies our process to construct Backend Idea Networks with heterogeneous data. In addition, it improves the efficiency of our system by treating each BIN independently.

Constructing good idea networks can be challenging for the unstructured textual information. Fortunately, many datasets have inherent relationships between nodes, such as hyperlinks, references, etc. These relationships between nodes provide an indication of potential connections. Thus, we construct a network with internal page links for Wikipedia and a network with phrase relations for ConceptNet.

However, there is no natural connections of tags and photos in MIRFLICKR. We use tag similarities to construct the network. Each image I_k in MIRFLICKR is associated with a set of tags $T_k = \{w_1, w_2, \ldots, w_k\}$, where w_j is a token in the vocabulary. We compute the similarity between two images I_i and I_j by computing a similarity score between their tags $Sim(T_i, T_j)$. Let $T_i = \{u_1, u_2, \ldots, u_m\}$ and $T_j = \{v_1, v_2, \ldots, v_n\}$. We extract C closest matches of u_i and v_j based on the Wu-Palmer similarities $WuP(u_i, v_j)$ between these two tokens [24], which calculates relatedness by considering the depths of the two synsets in the WordNet [17] taxonomies. Thus, $Sim(T_i, T_j)$ is computed by a mean of WuP scores of the matched token pairs. $Sim(T_i, T_j) = \frac{1}{C}\sum_{i=1}^{C} WuP(u^{(i)}, v^{(i)})$, where $u^{(i)} \in T_i$, $v^{(i)} \in T_j$, and $(u^{(i)}, v^{(i)})$ is i-th closest match. Note that, if fewer than C matches are found, we simply set the similarities of all unmatched pairs to 0. We therefore build an edge between I_i and I_j if $Sim(T_i, T_j)$ exceeds a threshold T.

5.2 Community Detection

Identifying communities in BIN is critical in our work to measure coverage in our objective function. The quality of communities is directly related to the recommendation result. We choose BigCLAM [25], a cutting-edge fast overlapping community detection algorithm for all our networks.

BigCLAM assumes dense overlaps among communities, and assumes that the probability of two given nodes having an edge is positively relavent to their number of common communities. This property is reasonalbe on all our networks, and thus can captures natural communities in our datasets. In Section 6 we talk about evaluating our community quality.

There exist another class of options to detect communities in our datasets, which are *topic models*, such as Latent Dirichlet Allocation (LDA), modeling topics using latent variables defined as distributions over the network. Topic models are good at detecting communities in thematic-based articles. In current phase, we prefer BigCLAM as a network model, where links can be utilized for recommendations.

5.3 Mapping PINs into BINs

In order to make suggestions from Backend Idea Networks, we first locate users' ideas from Problem Idea Network to our Backend Idea Networks. Once this mapping is done, we are able to obtain a standing point in Backend Idea Networks to find relevant ideas. Finding an optimal mapping is non-trivial, and here we discuss several approaches that we apply in our system.

5.3.1 Inverted Index

In order to achieve this mapping, we can use a direct inverted index where we search keywords in users' ideas on indexed BINs, and retrieve a ranked list of matched nodes with respect to the query (idea). We can use TF-IDF to assign weight to each single word in the query.

5.3.2 Query Expansion

Furthermore, we utilize WordNet [16] to expand the query with words from close synsets. Query expansion helps us make related suggestions when the input query is not directly observed in our idea network. For example, users' idea "Chihuahua" can be expanded to "dog" in its synset, thus improving the results of retrieval. We implement inverted index with query expansion on our image network.

5.3.3 Search Engine

Search engine is a sophisticated way to map a query into several possible candidates. We tried to build a search engine based on Lucene [11] to implement this mapping process, but for scale and simplicity reasons, in our system implementation we choose to make use of existing search engines. For the Wikipedia network, We use *Bing* to retrieve a set of Wikipedia pages related to user's idea, and choose a highest-ranked node (page) within our network.

5.3.4 Improvements on Mapping Assumptions

Mapping from PIN to BIN can be a challenging problem. Now our system is based on the assumption that nodes in PIN can be mapped and represented by one and only one node in BIN, while this is not necessarily true. Our current approach to relax this assumption is to introduce some randomness in the mapping.

In the future, we want to develop a probabilistic model where one node in PIN is mapped into multiple nodes with probabilities. With a random walk or inference model on top of this, we might be able to further improve the quality of mapping and recommendation.

5.4 Recommender: Optimization Algorithm

Finally we design an algorithm to retrieve K nodes that, given a set of parameters including *Idea Quality* (node score), *Community Relevance* (community score), and coefficients α, β, maximize our objective function, which is a linear combination of *Coverage* and *Combination* measures described in Section 4.

In [22], it is shown that *Cover* function is submodular for all nodes at all time. However *Comb* function is not submodular. In that case, we might not guarantee a greedy-hill-climbing algorithm to achieve $(1-1/e)$ *OPT* performance.

We note that there are two cases that a node is added into PIN: (1) a new idea is queried by users, and mapped into PIN; (2) a set of suggested ideas in BIN is returned, and thus added to PIN. In a basic use case of our system, users will generate some ideas, and ask for a set of suggestions, and re-generate ideas themselves. Here we aim to optimize the performance of (2) by a greedy algorithm.

When asked to return K nodes, the *OPT* solution which guarantees optimization, will enumerate all possible K combinations of nodes in BIN. Denote N as the size of BIN, the complexity of *OPT* is $O(N^K)$. A heuristic solution *H1* takes this task as suggesting K independent nodes and optimize each individual suggestion. Although this cannot guarantee global optimum, it is faster. For each iteration, $H1$ enumerates all neighbors of *PIN*, and find one node v. Then it will re- enumerate all neighbors of $PIN \cup v$ when suggesting another node, and this will go for K iterations. If the average degree of nodes in the network is D, and M is the size of *PIN*, then the complexity will be approximately $O(KMD)$: in each of K iterations, examine about MD nodes. Note that $N >> M, K, D$, so this is far better than the *OPT* method.

Here we propose a more efficient greedy optimization $H2$:

- Starting from $X = PIN$, in the first iteration, we examined all neighbors of X to find one node v that maximizes the objective.
- Then we update $X = X \cup v$, and maintain a list L of K nodes among all the neighbors examined in this iteration.
- At every other iteration, denote v' is the node added in the last iteration. Denote v's neighbors as $Nbr(v)$. In this iteration we only enumerate all nodes in list $Nbr(v) \cup L$, to find the next node to suggest.

This algorithm has complexity $O(MD + K(D+K)) = O(MD + KD + K^2)$, where M is the number of PIN, D is average degree in BIN, K is numbers of suggestions. In common and worst cases, $M, D >> K$, and this is K times faster than *H1*.

Figure 2 shows a counterexample where both $H2$ and $H1$ can be very bad. However in most cases, greedy algorithms do a good job, and we will evaluate the performance in future work.

5.5 Future Improvements

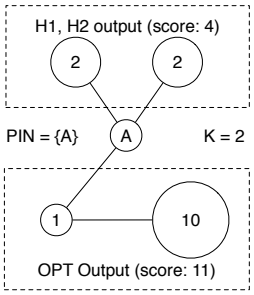


Figure 2: Greedy algorithms may have bad performance

Table 1: Dataset Statistics

Data type	Dataset	Nodes	Edges
Basic concepts	ConceptNet	4303083	10426726
Labeled images	MIRFLICKR	25000	404243
Encyclopedia	DBPedia English	9743	261920

In the future, we will try to find a better algorithm that gets a better estimation of the optimal solution, as well as being efficient enough.

Another direction to improve our algorithm is to enable adaptive learning. By interactions with users we are able to learn the $IdeaQuality$ and $CommunityRelevance$. Most simply, we can make use of the front-end feedback buttons ("like" or "dislike" an idea), which are currently implemented in our system, but just used for evaluation. In the future we can use the feedback for dynamic adjustments of scoring functions. Furthermore, we can apply learning algorithms on users' behaviors in using our system, to adjust objectives locally and globally.

6. EVALUATION

Evaluation of information systems often focuses on some standard metrics, e.g. accuracy, precision and recall. However, ground truth labels are difficult to obtain in case of ideation, which is a process involving human subjectivity. Standard methods fail to apply in our problem. In addition, ideation involves large amount of human interaction, the evaluation of our system demands feedback from human users. Thus, we conduct user studies to capture the effectiveness of our system in real-world brainstorming scenario.

In this section, we present an overview of our system and show evaluation results of our system.

6.1 System Overview

6.1.1 Network Construction

We build three Backend Idea Networks on three datasets, including basic concepts, Wikipedia and images. Table 1 shows the statistics of the networks constructed on these datasets.

The full MIRFLICKR network is used in our system. DBpedia network is subsampled to reduce the number of edges to a managable scale: we only select a set of the most-visited pages since Nov 2013 from the dataset [1].

For ConceptNet, the number of nodes ($\sim$4M) and edges ($\sim$10M) exceeds our computing power, and it is hard to subsample without losing information quality. Therefore, we complete the mapping with ConceptNet APIs. We use the APIs to retrieve concept relations and compute *Comb* score. The *Cover* score is set to a constant.

Table 2: Community Detection Result

Network	Communities	Likelihood
Image	9	-4.28×10^7
	100	-3.25×10^7
	300	-2.63×10^7
	500	-2.35×10^7
	1000	-1.98×10^7
	2500	-1.47×10^7
Wikipedia	100	-1.69×10^7
	150	-1.37×10^7
	200	-1.38×10^7
	300	-1.41×10^7
	974	-1.03×10^7

6.1.2 Community Detection

Community detection is the current bottleneck that constraints the size of our networks. We run BigCLAM algorithm on three networks with moderate size (∼25,000 nodes and ∼500,000 edges).

BigCLAM works in two phases: estimating number of communities with a cross-validation, and optimizing classification using MLE. Since the first step is slow, instead of finding the optimal number of communities, we sampled several numbers and picked ones that maximizes the likelihood. In Table 2 we list the community likelihood with respect to the number of communities. Currently we pick 2500 communities for image network (MIRFLICKR), and 974 for Wikipedia network, where in both networks the number of communities is $1/10$ to the number of nodes.

6.2 System Parameters

We implement KALEIDOSCOPE[2], an integrated backend idea recommender system and a frontend interface. As shown in Figure 1, the system loads three BINs with community information and inverted index in initialization phase. Each server maintains multiple PINs for different users. Users put their ideas into the search bar and get suggested ideas in forms of images, concepts, and encyclopedia documents. Figure 3 is a screenshot for the system frontend interface.

We tune the model parameters and select a set of parameters that give the best results. Specifically, we choose $\alpha = 1$, $\beta = 3$, $\lambda_w = 1$ for all communities, $cover_{idea}(w) = 0.2$.

We first set *Idea Quality* q_i to a constant. Our algorithm tend to suggest ideas according to existing suggested idea. Since user's ideas are fewer than suggested ideas, soon suggested ideas will dominate the topic and lead to a digression. Thus, we set different *Idea Quality* weights to user's ideas mapped to BIN (1.0), and suggested ideas (0.1). To make the suggestions reflect users' recent ideas, we assign a weight decay for users' ideas. Let their qualities be 2.0 in the beginning, the value get multiplied by a decaying factor of 0.8 after each brianstorming iteration.

Besides, different communities should not have the same *Community Relevance* (λ_w). In the future, we plan to allow users to specify their topics or learn personalized Community Relevance values from their ideas.

6.3 Evaluation

6.3.1 Methodology

[2]The system is available at `http://www.zifeishan.org/kaleidoscope/`, with released source code and data.

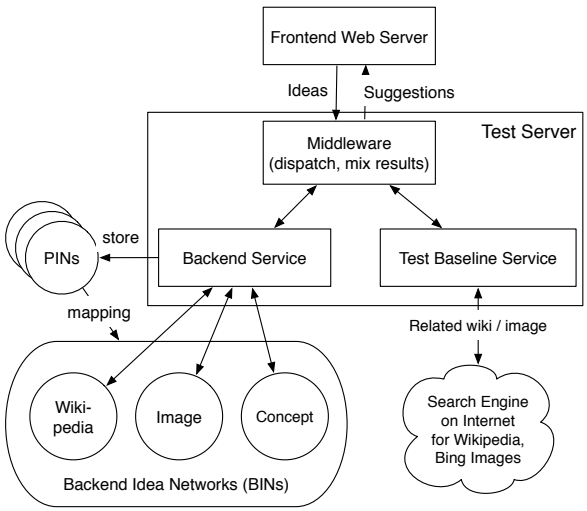


Figure 4: Blind-test System Architecture

We design a blind test to evaluate the quality of suggested ideas across different networks, compared to baselines of "most relevant ideas" given by search engines. We integrate the evaluation into our system, by implementing a user interface with feedback buttons connected to backend analytics. The architecture of our blind-test system is shown in Figure 4.

We give the following **brainstorming problems** to subjects who are Stanford students that have a knowledge of brainstorming. We try to design these questions with a large variety, covering different fields and potential thinking pattens:

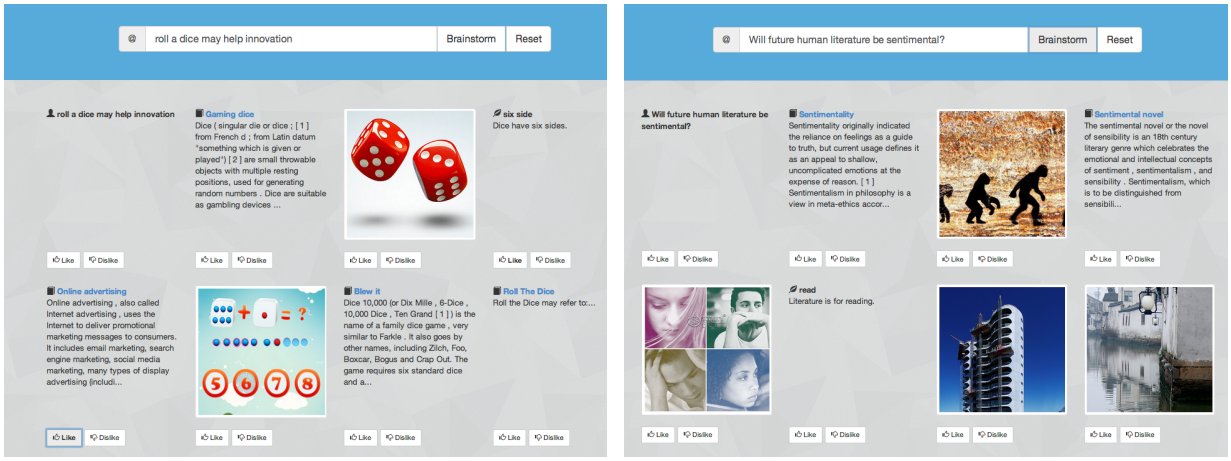
1. Design features for your favorite online application.
2. Think about things to do in next quarter.
3. Design a new way of transportation in Stanford.
4. What would be your most favourate restaurant?
5. Imagine your ideal significant other after 10 years.
6. Describe human literature after 50 years.
7. Imagine a university after 30 years.

Specifically, the suggestions from our system (*Backend Service*) and the *baseline systems* are displayed in the same appearance, equal in number, and randomly shuffled in order. We guarantee that users cannot tell the visual difference between them. The users are asked to provide "like" feedback to suggestions that *help or stimulate brainstorming*, and "dislike" feedback to suggestions that *distract users from brainstorming*, and ignoring a suggestion can be seen as a neutral feedback. We select the query results of users' current ideas in Wikipedia and Bing Image search as the baseline systems. Note that the sizes of our basline systems are significantly larger than the datasets in our system. We indexed 25,000 images and 9,743 Wikipedia articles, while the baselines have the entire set of Bing images and Wikipedia articles on the web.

According to the result of the test collected by the server, we compare the effectiveness between *different kinds of* suggestions as well as between *suggestions by our system and by the baseline*.

6.3.2 Results

The result of our blind tests is shown in Table 3. In the table, "Suggested Image", "Suggested Concept" and "Suggested Wiki" are



(a) (b)

Figure 3: Frontend Interface of Kaleidoscope System

Table 3: Evaluation Results

Type	Like	Dislike	Neutral
Suggested Image	38	14	44
Related Wiki	30	8	46
Related Bing Image	29	13	52
Suggested Concept	26	8	50
Suggested Wiki	20	15	63

computed in our backend system within our dataset; while, "Related Wiki" and "Related Bing Image" is retrieved from search engines, by directly searching the user's current idea. Figure 3 is snapshots of the blind-test interface, where suggestions are randomly mixed.

According to the results we see that generally, we achieve comparable results with the baseline. Two interesting comparisons are:

- Suggested images are more insightful than related images retrieved from Internet.
- Suggested Wikis are not as useful as related Wiki.

We hypothesize that there might be different kinds of status in computer-assisted ideation: one is *seeking for knowledge* and the other is *seeking for inspiration*. For a new topic that the user is less familiar with, they would like to look at related Wikipedia pages to have a big picture of the topic. However, in a topic that users are familiar with, they tend to be more inspired by our suggested images, most of which are not directly relevant, but often help users break through to new ideas.

In the future, we are interested in the following questions: with our growing *Problem Idea Network*, how to predict whether users have a certain piece of knowledge? How to tackle the suggestion problem with a probabilistic approach? How to generalize it to a knowledge visualization problem, and make use of users' knowledge structure?

Besides, we ask for general feedback from people attending the test. Over 80% of people feel that our system is helpful for brainstorming. We analyze the user-logs and find that in over 90% cases, users "like" at least one suggestion.

We can see that, although our system currently do not beat the baseline of related suggestions in all aspects, the results are promising: (1) the overall system is helpful to users, and we can also integrate the baseline as a way of suggestion; (2) we might have a better approach to model users' knowledge and preference; (3) with larger datasets, it is likely to get better results; (4) with fine-tuned parameters for idea quality, community relevance, etc., we have a large space for improvement.

6.4 Future Evaluation

Besides the evaluation we conduct, there are other ways to evaluate the system. We want to quantify how much our system helps users by measuring users' productiveness using our system. Specifically, there are some common measures for human-brainstorming process [14]: (1) fluency — how many ideas are generated per unit of time; (2) flexibility — how different those ideas are from what most people think up. We measure how fast and broad *users* generate ideas with or without the aid of auto-suggestions. Second and most straightforward, we ask questions to users, such as "do you like our system?", "do you find it helpful to have idea suggestions?", "which suggested ideas do you like best?" By a well-designed survey, we can better understand users' demands and re-design our objective measures.

Besides, we want to conduct modular tests on different system components: "are the links in networks reasonable?", "are the communities meaningful?", "is the mapping from PIN to BIN accurate and favorable by users?"

In the future, we want to invite more people, especially ideation experts (e.g. professors and students in *Stanford d.school*) to test our systems and provide useful feedback.

7. FUTURE WORK

In the future, we plan to craft our system into a public ideation platform that interactively present ideas to users in an ideation process. We propose to achieve it by deploying our algorithms into a real-time online brainstorming system named Sparkl [8]. We plan to add automatic idea suggestion modules into Sparkl.

Meanwhile, it is exciting to combine Problem Idea Networks together as another Backend Idea Network. This allows us to store all users' ideas into a uniform network, and allows brainstormers to see each others' ideas. A collaborative, crowdsourcing-like knowledge base is our long-term goal.

8. CONCLUSION

In this paper, we introduce KALEIDOSCOPE, our working system as an attempt to achieve interactive computer-assisted idea generation. We propose a model to capture human brainstorming and allow computers to collaborate with people and take part in innovation.

We define the *idea network* model: a *Problem Idea Network (PIN)* captures all the ideas in a ideation process, and *Backend Idea Networks (BIN)* include all ideas in the knowledge bases our system hold. We craft a "3C" objective function of idea suggestion, including a high *coverage* on different fields (communities), and a good *combination* of existing ideas, with a constraint of *connection*.

Using our deployed system, we conduct a system evaluation based on user surveys, where we get promising results as a working system. We aim at deploying KALEIDOSCOPE as an online ideation platform, and exploring ways to aggregate PINs as a collaborative knowledge base.

To the best of our knowledge, our work is the first attempt to enable human-computer collaboration in an interactive ideation process. Our contribution lies in the following: (1) we initiate the methodology of computer-assisted ideation; (2) we craft objectives and algorithms that shed insights on recommending knowledge in information networks; (3) we build a system to facilitate ideation with heterogeneous knowledge data and evaluate its effectiveness.

Acknolwegement

The authors would like to acknowledge Professor Jure Leskovec for critical advice and encouragements about this project idea. Thanks to all the Stanford students who attended the evaluation. Thanks to Jaewon Yang for helping us in the use of BigCLAM community detection tool. Thanks to Doctor Dafna Shahaf for the useful initial advise and following discussions about the project. Thanks to Justin Cheng for great suggestions for our proposal and milestone report. Thanks Justin for being excited about this project, and looking forward to more advice. Thanks to Manwen Guo and Scott Cheng, who have built the Sparkl system, provided us platform support, discussed with us many times about market needs of this project to help our prototyping.

9. REFERENCES

[1] Page view statistics for wikimedia projects. `http://dumps.wikimedia.org/other/pagecounts-raw/`.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[3] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *International Joint Conferences on Artificial Intelligence*, pages 2670–2676, 2007.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May, 2011.

[5] K. Bollacker, P. Tufts, T. Pierce, and R. Cook. A platform for scalable, collaborative, structured information integration. In *International Workshop on Information Integration on the Web*, 2007.

[6] H. M. Cannon, P. G. Carroll, and B. L. Seamons. Using the ideafisher idea generation system as a decision support system in marketing strategy courses. *Developments in Business Simulation and Experiential Learning*, 20:27–30, 1993.

[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. H. Jr., and T. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313. AAAI Press, 2010.

[8] S. Cheng and M. Guo. Sparkl: Empower creative thinking. `http://www.sparkl.us/`.

[9] O. Etzioni, M. Cafarella, D. Downey, A. maria Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *ARTIFICIAL INTELLIGENCE*, 165:91–134, 2005.

[10] Freebase. Freebase: A community-curated database of well-known people, places, and things. 2007. `http://www.freebase.com`.

[11] E. Hatcher, O. Gospodnetic, and M. McCandless. Lucene in action, 2004.

[12] M. Hilbert and P. López. The world's technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, 2011.

[13] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2008. ACM.

[14] T. C.-C. Institute. Brainstorming. `http://www.cocreativity.com/handouts/brainstr.pdf`.

[15] H. Liu and P. Singh. Conceptnet: A practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, 2004.

[16] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[17] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[18] NELL. Nell: Never-ending language learning. 2010. `http://rtw.ml.cmu.edu/rtw/`.

[19] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In T. Walsh, editor, *IJCAI*, pages 2734–2739. IJCAI/AAAI, 2011.

[20] D. Shahaf, C. Guestrin, and E. Horvitz. Metro maps of science. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1122–1130, New York, NY, USA, 2012. ACM.

[21] D. Shahaf, C. Guestrin, and E. Horvitz. Trains of thought: generating information maps. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *WWW*, pages 899–908. ACM, 2012.

[22] D. Shahaf, C. Guestrin, and E. Horvitz. Trains of thought: Generating information maps. In *Proceedings of the 21st international conference on World Wide Web*, pages 899–908. ACM, 2012.

[23] Wikipedia. Wikipedia: The free encyclopedia. 2007. `http://www.wikipedia.org/`.

[24] Z. Wu and M. S. Palmer. Verb semantics and lexical selection. In J. Pustejovsky, editor, *ACL*, pages 133–138. Morgan Kaufmann Publishers / ACL, 1994.

[25] J. Yang and J. Leskovec. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.

[26] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: open information extraction on the web. In *NAACL '07*, pages 25–26, Morristown, NJ, USA, 2007. Association for Computational Linguistics.