Chengwei Zhang
Department of Mechanical Engineering,
Tsinghua University,
A1003-1 Lizhaoji,
Beijing 100084, China;
Department of Mechanical Engineering,
University of California, Berkeley,
Berkeley, CA 94720
e-mail: zhangcw13@mails.tsinghua.edu.cn

Youngwook Paul Kwon
Department of Mechanical Engineering,
University of California, Berkeley,
2114 Etcheverry,
Berkeley, CA 94720
e-mail: young@berkeley.edu

Julia Kramer
Department of Mechanical Engineering,
University of California, Berkeley,
354/360 Hearst Memorial Mining Building,
Berkeley, CA 94720
e-mail: j.kramer@berkeley.edu

Euiyoung Kim[1]
Mem. ASME
Jacobs Institute for Design Innovation,
University of California, Berkeley,
2530 Ridge Road,
Berkeley, CA 94720
e-mail: euiyoungkim@berkeley.edu

Alice M. Agogino
Fellow ASME
Department of Mechanical Engineering,
University of California, Berkeley,
415 Sutardja Dai Hall,
Berkeley, CA 94720
e-mail: agogino@berkeley.edu

Concept Clustering in Design Teams: A Comparison of Human and Machine Clustering

Concept clustering is an important element of the product development process. The process of reviewing multiple concepts provides a means of communicating concepts developed by individual team members and by the team as a whole. Clustering, however, can also require arduous iterations and the resulting clusters may not always be useful to the team. In this paper, we present a machine learning approach on natural language descriptions of concepts that enables an automatic means of clustering. Using data from over 1000 concepts generated by student teams in a graduate new product development class, we provide a comparison between the concept clustering performed manually by the student teams and the work automated by a machine learning algorithm. The goal of our machine learning tool is to support design teams in identifying possible areas of "over-clustering" and/or "under-clustering" in order to enhance divergent concept generation processes. [DOI: 10.1115/1.4037478]

Keywords: concept clustering, design process, machine learning, word embedding

1 Introduction

1.1 Concept Classification, Clustering, and Generation in the Design Process. Classification and clustering are semantically similar in that they are both tasks of grouping a given set of objects into meaningful and useful groups. However, there are important differences: classification is the task of grouping objects into predefined classes created with prior knowledge, whereas clustering is the task of finding underlying patterns and grouping objects together based on their similarity [1]. Concept clustering is a crucial part of the product development process as it allows designers to interpret the concepts they generate and decide which of these concepts to develop, modify, or adopt [2]. The process of reviewing multiple concepts provides a means of communicating concepts developed by individual team members and by the team as a whole.

Concept classification and clustering have a long history rooted in psychology [3] and artificial intelligence. Smith [4] defines a "concept" as "a mental representation of a class or individual." Dong and Agogino [5] developed a learning algorithm to automate the process to handle a large quantity of natural language texts to construct these design representations or concepts. Their attempts recommend that designers find relevant information based on terminologies and organize the data in a more meaningful way. This approach is similar to Wood et al. [6] in text-based information analysis, and our research was motivated by such text data/information retrievals for concept clustering in new product development.

In the design process, choosing the right concept classes requires a significant amount of time and effort to narrow down concepts into a manageably small number of compelling clusters; it is crucial to explore these clustered opportunity areas before moving on to the next phase of the product development process, such as concept selection [7–9]. Concept clustering can also be used to identify cluster areas that have a relatively small number of concepts, which can indicate to the design team useful targets for further divergent concept generation.

Clustering of design concepts is useful for organizing concepts into similar groups, therefore aiding designers in removing duplicate concepts and managing a large number of similar concepts [1]. Clustering, however, can also require arduous and ineffective iterations, and the resulting clusters may not always be effective for team communication, which is key to the success of the design process [10]. While, in machine learning, conceptual clustering

[1]Corresponding author.

Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received February 20, 2017; final manuscript received July 18, 2017; published online October 2, 2017. Assoc. Editor: Harrison M. Kim.

methods have been well developed as a means to summarize and organize data [11], the application of concept clustering for designers who are actually using it in real-world design projects during concept generation has been relatively underexplored.

From a machine learning perspective, classification is often performed with supervised learning against a known standard, whereas clustering typically uses unsupervised learning as there is rarely a standard to learn against. Many clustering algorithms have been proposed, including K-means [12], spectral clustering [13], mean shift [14], and hierarchical clustering [15].

In our research, in order to automatically generate concept clusters, we compute the numerical similarity between concepts through paragraph similarity with word embedding as a building block and compare the results between automatic and human clustering.

1.2 Semantic Clustering. Semantic clustering has been studied widely and many approaches have been proposed. In our review, we discuss semantic clustering in terms of three parts: (1) word embedding, (2) clustering algorithms, and (3) applications.

Semantic clustering is a form of computational linguistics that evaluates quantitative measures of similarities in the meaning of words. In order to process human languages computationally, there have been approaches to embed, or encode, "words" into vectors [16–18], as is done with "pixels" of images or "sampled frequencies" of audio data. There is no straightforward way to conduct the embedding, but the main idea is that words of semantically similar meaning should have similar vectors (i.e., nearby points in the high-dimensional embedding space).

Overall, there are two branches in word embedding: counting-based methods and predicting-based methods. Counting-based methods (e.g., latent semantic analysis) basically rely on the statistics of word appearance in documents. Predicting-based methods (e.g., neural language models) try to learn embedding by predicting a word from the nearby words. More details on these two branches can be found in Baroni et al. [19].

Deep learning uses computational models with multiple processing layers to learn representations of data with multiple levels of abstraction [20]. Given the recent success of deep learning in various other fields, there have been several approaches using deep learning in word embedding. For example, Word2Vec [17] is known as a well-defined and useful predicting-based method of word embedding. In this research, we use Word2Vec as our building block.

In both counting-based and predicting-based methods, there are various ways to handle "textual data" given by a set of words (i.e., word vectors) for the purpose of clustering them. Broadly speaking, there are hierarchical approaches (e.g., non-negative matrix factorization [21]) and partitional approaches (e.g., K-means [22]). There is no well-established preference between the two approaches. While Steinbach et al. [23] posit that K-means performs better than hierarchical clustering in the document clustering domain, Zhao and Karypis [24] state that even though partitional clustering algorithms (including K-means) are computationally efficient for large datasets, they are inferior to agglomerative hierarchical methods in terms of clustering quality. In this research, cosine similarity measures rather than Euclidean distance made more sense, thus favoring hierarchical clustering over partitional clustering.

Le and Mikolov [25] introduced Para2Vec, which is an extended version of Word2Vec, assigning high-dimensional vectors to paragraphs. However, the Para2Vec algorithm was originally designed to handle a very high number of paragraphs, and thus was less useful for our relatively small number of concept descriptions (~1000), given as short text paragraphs. Therefore, we define our own similarity metric for short texts, as discussed in Sec. 2.2.

Semantic clustering has seen a wide range of applications, including document classification [26–28] filtering repetitive

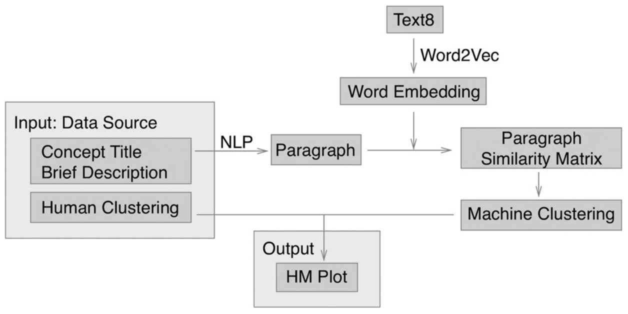


Fig. 1 Processing flow of our proposed method

news/blog feeds [29], identifying topics in programming source code [30], identifying structure in a patent database [31], and identifying distance of designer's points of view [32]. In this research, we use semantic clustering on short text descriptions, which has previously been shown to be successful in mobile malware app detection [33], customer review classification [34], and Twitter information filtering [35]. We apply our semantic clustering algorithm in an effort to support design processes, which was an analogous focus in Dong and Agogino's [36] work.

2 Research Design

As shown in Fig. 1, we use the concept clustering data collected from a graduate level design course. Taking certain fields of them as paragraphs, a similarity matrix is calculated by the word embedding generated from a general corpus. Then, concepts are clustered by machine learning algorithms. We compare the machine clustering with human clustering in the form of human–machine plots (HM plot).

2.1 Data Source. The concepts we use as input data in this work were collected from a human-centered design course we will label as ME300. The course was taught in Fall 2016 at the University of California, Berkeley, and aimed to teach human-centered design methods using theDesignExchange website.[2] Students in ME300 were split into teams, each tasked with a unique design challenge (Table 1). During the semester, student teams generated and described 1154 concepts using a "half-sheet" template (Fig. 2) that included the concept's title, a brief text description, a list of key attributes/features, a rough sketch of the idea, and a list of creativity methods used (if any). The teaching team provided the half-sheet template to students as part of an individual homework assignment. The students were asked to generate ideas individually and bring them to class to share with their team and to be used in a concept clustering activity. While concept clustering was part of the class's team assignments, three (out of 14) teams did not include the clustering activity results in their documentation. Therefore, we focused on the 11 teams, renumbered as team 1–11, who explicitly documented their clusters across a complete list of all generated concepts in their submissions. The natural language toolkit (NLTK),[3] was used to mark the speech of tag in concept description. As shown in Table 2, each team described their concepts in 17.77 words, on average, including 5.97 nouns and 3.47 verbs.

In this research, we aim to address the primary research question as to how the results of machine learning-based concept clustering differ from manual concept clustering. We then explore implications for supplementing human clustering with machine learning clustering in product design teams.

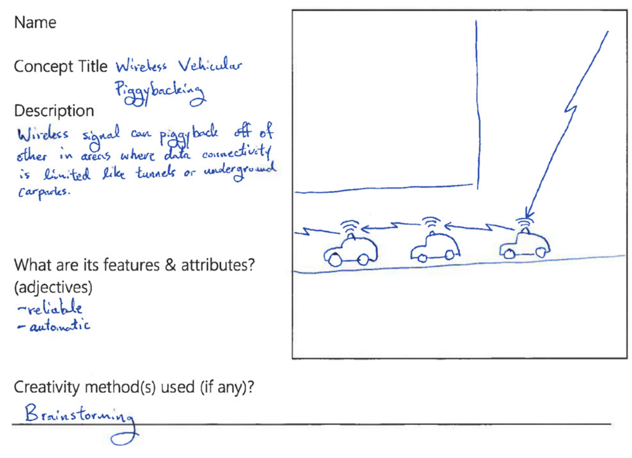
2.2 Machine Clustering Using Machine Learning Methods. Our goal is to quantitatively cluster design concepts, which are in the format of natural language paragraphs written by

[2]https://www.thedesignexchange.org
[3]http://www.nltk.org

Table 1 ME300 teams and their associated projects

Team no.	Team design challenge
1	Understanding the interface between humans and autonomous vehicles
2	Navigating the deep seas with an autonomous underwater vehicle
*	Connecting kids and parents using wearable technology
3	Making the cycling experience safer more navigable using wearable technology
*	Automatically generating control-based dynamic simulation models
4	Creating and implementing esthetic wind turbines
5	Improving the delivery of ear medication
6	Using microneedle arrays for transdermal drug delivery
7	Integrating innovative robotics with educational curricula
*	Including nature in the search for ideal housing
8	Improving the outcomes of spinal surgeries
9	Improving disaster relief with free-standing robotics
10	Exploring a new way to harvest ocean power
11	Enhancing theme park experiences through improved character costumes

Note: Teams with * did not sufficiently complete their concept clustering exercise and therefore we do not include their data in our analysis.



Name

Concept Title Wireless Vehicular Piggybacking

Description
Wireless signal can piggyback off of other in areas where data connectivity is limited like tunnels or underground carparks.

What are its features & attributes? (adjectives)
- reliable
- automatic

Creativity method(s) used (if any)?
Brainstorming

Fig. 2 An example half-sheet from team 1

members of product design teams. Automatic clustering requires a similarity (or distance) metric between items. For example, consider a pair of concept paragraphs in our data set from team 1's autonomous vehicle project: c_1 and c_2 (c_1 = "instead of using steering wheel, user will use screen to control car" and c_2 = "user can sleep inside a moving car"). As the concept description is a relatively short paragraph, we applied semantic similarity, rather than word similarity, to avoid sensitivity to word choice. Then, we propose a paragraph similarity metric $\mathrm{sim}(c_1,\ c_2)$ based on semantic similarity.

2.2.1 Word Similarity. As a first step, we conducted word embedding using Word2Vec, which means we assigned a d-dimensional continuous vector to each word. We used $d = 200$ as typical [37], applied to 100 MB text8[4] as the corpus, including 47,134 unique words in the vocabularies. Text8 was cleaned to contain only letters *a*–*z* and nonconsecutive spaces. Using the Tex8 corpus, a word can be expressed as a d-dimensional vector $\mathbf{w}$. For a pair of words, $\mathbf{w}_1$ and $\mathbf{w}_2$, we use similarity metric such as:

[4]https://cs.fit.edu/~mmahoney/compression/textdata.html

Table 2 Statistical details of concept description

Team no.	Average description length	Average # nouns	Average # verbs
1	15.95	5.89	3.06
2	24.84	6.66	4.84
3	13.33	5.24	3.01
4	24.11	7.59	4.32
5	17.43	5.33	3.49
6	21.93	6.75	4.25
7	7.86	4.81	1.61
8	18.05	5.98	3.57
9	15.74	5.34	3.04
10	13.42	5.22	2.84
11	22.80	6.90	4.18
Average	17.77	5.97	3.47

$$\mathrm{sim}(\mathbf{w}_1,\ \mathbf{w}_2) = \frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{\mathbf{w}_1 \mathbf{w}_2} \tag{1}$$

To show the efficacy of our word embedding, we illustrate similarity calculations for team 1 between the query words and k nearest neighbor words. In Table 3, we show four nearest words for a given query word. From this table, we see that the nearest neighbor words are, in fact, semantically similar to the query words. Therefore, we proceed with our word similarity metric.

2.2.2 Paragraph Similarity. We define the concept, together with its description, as a paragraph. A paragraph is a set of featured words, $c_i = \{\mathbf{w}^i_1,\ \mathbf{w}^i_2,\ \ldots,\ \mathbf{w}^i_{l_i}\}$, where the featured words refer to the noun and verb words in the paragraph, and l_i is the number of featured words in c_i. The number of words in each paragraph, or each concept, varies.

For a given pair of concept paragraphs, c_i and c_j, we define a similarity matrix $\mathbf{S}_{c_i \times c_j} \in \mathbb{R}^{\{l_i \times l_j\}}$ with pairwise word similarities such as

$$\mathbf{S}_{c_i \times c_j}|_{(m,n)} = \mathrm{sim}(\mathbf{w}^i_m, \mathbf{w}^j_n) \tag{2}$$

where $\mathbf{S}|_{(m,n)}$ represents (m, n) element of the matrix $\mathbf{S}$. Upon this full pairwise matrix, we calculate real-valued concept paragraph similarity score sim (c_i, c_j) by averaging p percent of highest scores in the matrix. Specifically, we denote

$$\mathrm{sim}(c_i,\ c_j) = 1/K \sum_{k=1}^{K} d_k \tag{3}$$

where d_k is the largest K scores in $\mathbf{S}_{c_i \times c_j}$ and $K = p \cdot l_i \cdot l_j$. The parameter p is a percentage that controls the extent to which the paragraph similarity calculation uses the words from the concept description. A too low p-value may mislead the algorithm to focus only on the words that are frequently used but may not be topic related. On the other hand, if the p-value is too high, words that are specific but isolating in the word vector space may mislead the algorithm to lower the similarity between a concept pair. Given these constraints, we empirically tested different p values and found $p = 15\%$ to be the most effective with this application. Note that the similarity score has a range between $[-1, 1]$.

To illustrate these algorithmic techniques, consider the example of team 1 again. Using the paragraph similarity metric, we calculate all pairwise distances between the 82 concepts generated by team 1. Team 1 focused on the human interactions with an autonomous vehicle, and their concepts are shown as a heat map [38] in Fig. 3. The concepts in the heat map have been sorted by their machine-derived semantic similarity, with similar concepts close

Table 3 Four nearest neighbor (NN) words to each query word in the embedding space. In each cell, we show the similarity calculation between the query word and the nearest neighbor word.

Query word	1NN	2NN	3NN	4NN
Steering	Loading (0.813)	Brake (0.806)	Hydraulic (0.784)	Crank (0.780)
Driving	Pulling (0.608)	Mounting (0.607)	Glider (0.600)	Charging (0.597)
Safety	Maintenance (0.765)	Handling (0.694)	Monitoring (0.691)	Surveillance (0.675)

to each other. Thus, the block-diagonal structure shows the algorithm's ability to cluster semantic relevant concepts. As shown in the magnified part of the heat map, concepts co-located in the high-similarity areas are semantically close to each other (e.g., "comfortable light in car," "sleeping inside car," "obstacle warning system," "sensor failure warning," "emergency warning," "low battery warning," and "maintenance warning"). These dark-colored areas visually distinguish the concept clusters that the algorithm generated.

2.2.3 Clustering. The similarity matrix allows us to conduct hierarchical clustering [15] of each team's concepts. Although the number of clusters can be adjusted, we choose—as a first step for now—to set the number of machine clusters to be the same as the number of human clusters, thus simplifying the comparison between them. In the future, we will continue to develop our algorithm to understand how it behaves when a different number of clusters are chosen or where the algorithm self-generates the optimal number.

To address the problem of naming each machine-generated cluster, we attempt to find out those words that contribute most when calculating the concept similarity matrix. Specifically, for the kth concept cluster, $cl_k = \{c_1, c_2, ..., c_{n_k}\}$, we count every word pair for every concept pair $(c_i,\ c_j)$ where $c_i, c_j \in cl_k$ and $i < j$, and choose the most frequent word pair as a cluster label. When a concept cluster includes only one concept, we choose its concept name as a cluster label.

3 Data Analysis and Comparisons

To compare the results of concept clustering done manually (human clustered) and automatically (machine clustered), we created human–machine plots (see Sec. 3.1). The HM plots show two distinct patterns across clustered concepts: under-clustering and over-clustering. Our success in identifying these patterns highlights a potential application area where deep learning tools can support design teams. We discuss these patterns and their implications in the rest of the paper. The remainder of the paper discusses the preliminary stages of our machine learning algorithm on natural language descriptions of concepts and illustrates clustering patterns and their implications.

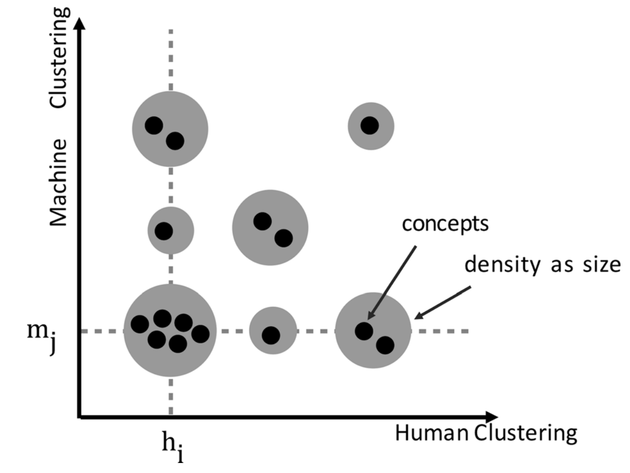


Fig. 4 A depiction of an HM plot. The *x*-axis shows the index of human clustering, the *y*-axis shows the index of machine clustering, and the size of the circles shows the number of concepts in each cluster.

3.1 Human–Machine Plot (HM Plot). The student teams manually assigned each concept to one of their human defined clusters, h_i, and our algorithm automatically assigned each concept paragraph to one of the machine learning based clusters, m_j. In Fig. 4, we illustrate an example distribution based on $(h_i,\ m_j)$ locations. We note two distinct patterns of concept clusters that emerge from the HM plot data: (pattern #1) under-clustering and (pattern #2) over-clustering (Fig. 5).

Pattern #1: Machine suggests that teams under-clustered their concepts.

Under-clustering refers to a pattern where the algorithm breaks apart a single large human-generated cluster into multiple clusters. The left portion of Fig. 5 depicts the under-clustering pattern of part of the clusters.

This pattern highlights an opportunity for the students to revisit their clusters and think more divergently in the new machine-generated cluster areas. When the clusters are too large, as is the

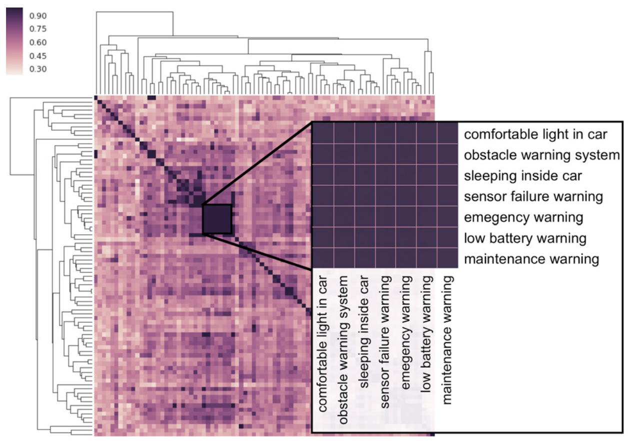


Fig. 3 Heat map of concepts generated by team 1

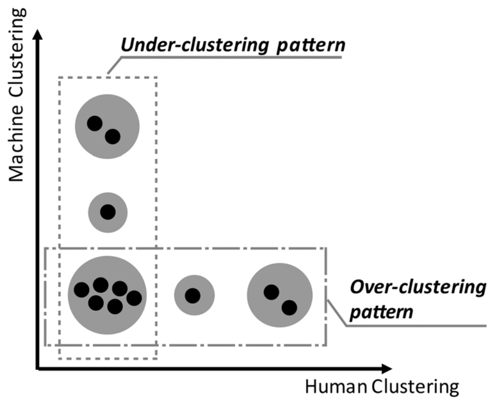


Fig. 5 A depiction of the under- and over-clustering pattern of part of the clusters on an HM plot

case in under-clustering, the teams may lose the main value proposition of that group of concepts. The cluster becomes a grab bag, lacking definition that is necessary for the cluster to be useful in the downstream selection process. To remedy this challenge, the student teams can see the clusters created by the algorithm and use this to reconsider their clusters, which may spur further divergent concept generation in each new cluster.

Pattern #2: Machine suggests that teams over-clustered their concepts.

Over-clustering refers to a pattern where the algorithm combines several small human-generated clusters into a single large machine cluster. The lower portion of Fig. 5 depicts the over-clustering pattern of part of the clusters.

This pattern highlights an opportunity for students to revisit their clusters and either perform concept generation on these minimally populated clusters, or think about combining similar clusters together. By pointing out opportunities to revisit over-clustered concepts, teams might consider expanding concept generation in underpopulated small clusters. Conversely, if the team believes the number of concepts is sufficient, they may want to pare down their clusters into sets that could become more useful in further convergent concept selection.

We constructed HM plots for all 11 teams that completed their own concept clustering (see Table 4).

In Secs. 4 and 5, we explore the patterns of under- and over-clustering for each team. While we do not claim that there is a "right" or "wrong" way to cluster concepts, nor do we believe that machine-generated clusters are better than human-generated clusters; rather, we posit that our algorithm provides meaningful opportunities for teams to revisit their clusters and thus engage in more concept generation and refinement processes.

4 Result and Human–Machine Plots Analysis

4.1 Concepts and Clusters Generated by Human Teams. Overall, the 11 teams we analyzed generated 930 concepts out of the 1154 concepts generated in the class (Table 4). On average, each team created 12.6 clusters of concepts. Team 7 created the most concepts, with 120 concepts generated. Team 6 generated the least (40). Team 3 created the most concept clusters (21) and team 5 created the least concept clusters (9). The number of concepts and clusters created by each team is shown in Table 4.

4.2 HM Plots for ME300 Teams. In Fig. 6, we show the overview of the HM plots for the 11 teams who sufficiently completed their concept clustering exercises. We highlight these HM plots here to show how various student teams exhibited patterns of both under-clustering and over-clustering.

Table 4 Number of concepts and clusters created by each team

Team no.	# team concepts generated	# clusters each team generated	Avg. # concepts per cluster
1	82	15	5.5
2	70	17	4.1
3	70	21	3.3
4	103	18	5.7
5	100	9	11.1
6	40	11	3.6
7	120	20	6.0
8	80	10	8.0
9	100	8[a]	20.4
10	59	5[a]	23.8
11	106	5[a]	22.2
Total	930	12.6	10.3

[a]Overlapped clusters means some concepts were put in more than one cluster.

Figure 6 shows that teams 1–8 exhibit a similar linear pattern of the location of circles that display the density of concepts in each cluster, where the human and machine clusters are grouped in a similar manner. Teams 9–11, however, show markedly different patterns, as they created overlapping clusters where a single concept was clustered into several different clusters.

For a given HM plot, we define three statistical terms: congruency, under-clustering index, and over-clustering index. The congruency (represented as a percentage) is the number of concepts on the diagonal of the plot divided by all the concepts generated by a team. It shows the similarity between machine clustering and human clustering. The under-cluster index is the maximum number of machine clusters whose concepts fall into one human cluster divided by the number of clusters for the team. Note that the machine clustering algorithm was set to have the same number of clusters as the team generated. The over-cluster index is the maximum number of human clusters whose concepts fall into one machine cluster divided by the number clusters for the team.

As an example, let us assume that a team generated 13 concepts and clustered them into three groups. Therefore, we would create an HM plot with three human clusters and three machine clusters. The number of concepts in the HM plot would simply be denoted as a 3×3 matrix where the rows are machine clusters and the columns are human clusters, e.g., $\begin{bmatrix} 0 & 1 & 2 \\ 0 & 3 & 0 \\ 5 & 2 & 0 \end{bmatrix}$. The 13 total concepts are spread across the human and machine clusters (i.e., $13 = 5 + 2 + 3 + 1 + 2$). In this case, we calculate congruency$=$(concepts in diagonal/total concepts)$=((5+3+2)/13)= 0.77$. To understand the degree of under-clustering and over-clustering, we calculate the under clustering index$=\max((1/3),(3/3),(1/3))=1$ (columnwise maximum spread-out), and over clustering index$=\max((2/3),(1/3),(2/3))=0.67$ (rowwise maximum spread-out).

Table 5 shows these statistical summaries of the eight teams' HM plots. In our data set, teams 9, 10, and 11 created overlapping clusters, meaning that some of their concepts were put in more than one cluster. If the teams grouped individual concept to multiple clusters, no diagonal patterns exist. Therefore, we excluded congruency, over-clustering, or under-clustering indices of the data from team 9-11 in this analysis.

The under-cluster column in Table 5 shows the name and size of the human cluster whose concepts were grouped into most machine clusters. These were the most under-clustered and were broken up into the most machine clusters.

The over-cluster column in Table 5 shows the name and size of the machine cluster whose concepts were grouped into the most human clusters. Teams 5 and 7 have high under-clustering indices, indicating that some human clusters in these teams are too big, and can be split into more segments. These two teams also have high over-clustering indices, indicating that some concepts share similar attributes but are put into different human clusters.

To give more details about the HM plot, we consider team 1 again as an example (Fig. 7). They focused on a project for understanding the interface between humans and autonomous vehicles, generated 82 concepts and divided them into 14 categories. They exhibited both patterns of under-clustering and over-clustering. Figure 7 shows the HM plot comparing the team-generated concept clusters and machine-generated concept clusters. The size of each bubble corresponds to the number of concepts within that cluster.

4.3 Example of Under-Clustering Patterns. Figure 8 shows an overview of the teams' example patterns in under-clustering their concepts. To illustrate the under-clustering in detail, we consider team 4 as an example. Team 4 focused on a project for creating and implementing esthetic wind turbines, generated 103 concepts and divided them into 18 clusters. Table 6 shows the cluster named "children friendly design" comprised of 12 concepts. Because the team under-clustered, the algorithm broke

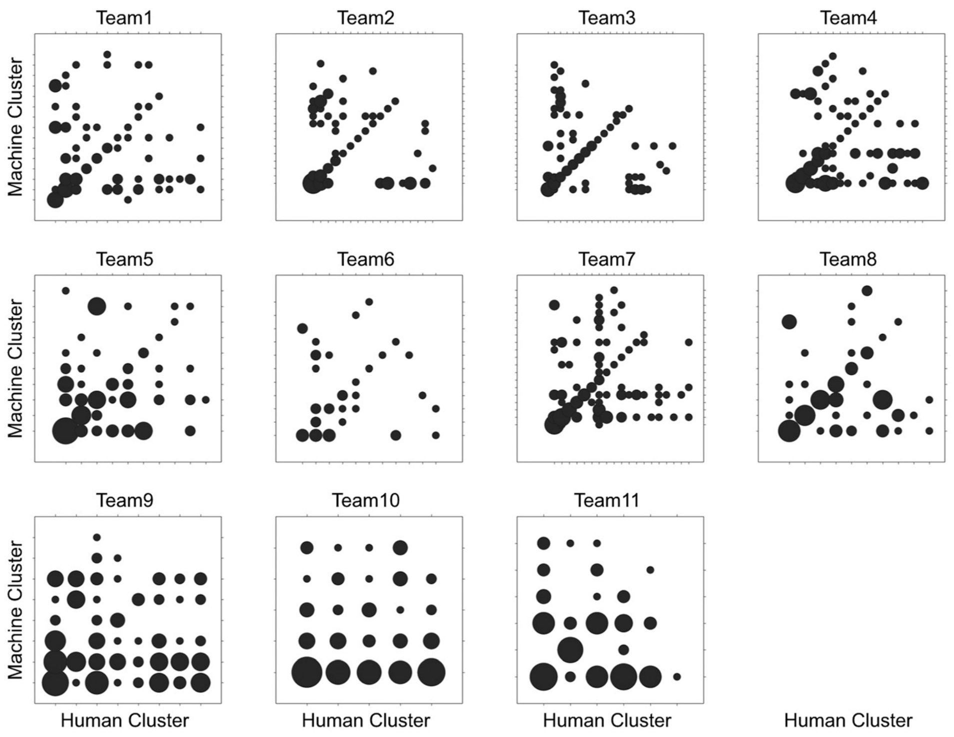


Fig. 6 Overview of 11 teams' HM plots. In the subplot, the *x*-axis shows the human clusters, the *y*-axis shows the machine clusters, and the size of the circles shows the number of concepts in each cluster. More description of the HM plots are available under the "Supplemental Materials" tab for this paper on the ASME Digital Collection.

Table 5 Summary of eight teams' HM plots: congruency, indices of over-clustering and under-clustering

Team no.	Congruency	Under-cluster index	Under-cluster title (size)	Over-cluster index	Over-cluster title (size)
1	0.29	0.47	Leisure and entertainment (10)	0.6	Car; driver (17)
2	0.34	0.35	Attachments (16)	0.47	Vehicle; equipment (25)
3	0.39	0.43	Outdoor (12)	0.43	Game; games (21)
4	0.27	0.37	Modern wind farms (9)	0.68	Wind; device (31)
5	0.36	0.6	Cool bottles (25)	0.8	Ear; ears (23)
6	0.38	0.36	Added feature (11)	0.73	Needles; micro (19)
7	0.29	0.74	Tensegri-home (22)	0.84	Robot; robots (42)
8	0.46	0.5	New implant (16)	0.7	Fracture; needle (20)

these concepts down into smaller clusters with several different labels. In the machine labels' view, the first four concepts in this cluster tend to be about activity, while the others emphasize wind devices and energy generators.

4.4 Example of Over-Clustering Patterns. Likewise, Fig. 9 shows an overview of the teams' example patterns in over-clustering their concepts. To illustrate the over-clustering in detail, we take team 3 as an example. Team 3, focused on a project for making the cycling experience safer and more navigable using wearable technology, generated 70 concepts and divided them into 21 clusters. The team clustered two concepts (#51 "hiking challenge app" and #56 "biking challenge app") into the two different clusters of "outside exercise" and "cycling," but the machine found that they are both related to app development (see Table 7). This relation happened to be mentioned in the concept description, and the machine succeeded in finding that pattern. This raises the question as to whether the common feature should be the implementation as an app; or should it focus on the different experiences associated with outdoor exercising, in general, versus cycling, in particular? Clearly, there is a divergent thinking opportunity to generate more concepts in any of these categories.

5 Discussion/Conclusion

In this paper, we attempt to build a framework for concept clustering in product development by applying a machine learning-based concept clustering tool on natural language descriptions of concepts developed by design teams. By comparing the machine-generated clusters to the human team-generated clusters, we see patterns of apparent over-clustering and under-clustering that may improve team performance.

Teams who under-cluster may have created too few clusters for their concepts, therefore resulting in large and unwieldy clusters. These teams might benefit by expanding their few clusters into smaller clusters that better represent the themes or functionality represented in the concepts. Our algorithm highlights areas where teams might consider creating more clusters in order to break apart their ideas into more specific and descriptive themes.

Teams who over-cluster, on the other hand, may have created too many clusters for their concepts when they could have used fewer clusters to give a more accurate representation of their

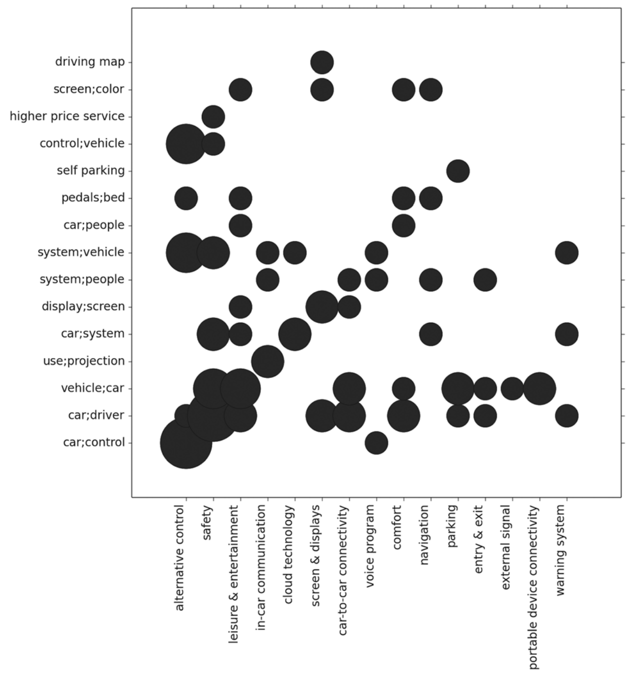


Fig. 7 Sample HM plot of team 1

concept themes. Thus, these teams might benefit from convergent thinking to pare down their many clusters into a more parsimonious cluster set. They might use concept merging to mix and match similar features, functionalities, or experiences. Over-clustering may also indicate opportunity spaces that would benefit from further concept generation within each cluster as well. In other words, teams might benefit from taking each of their small conceptual clusters and generating more concepts in systematic ways by expanding the concepts associated with each of these clusters.

6 Future Research

This paper outlines preliminary research in developing a tool that identifies meaningful patterns in clustering of design concepts. Our goal is to develop a machine-learning tool that will assist in mediating communication [10] among design team members and help them focus in areas that might benefit from further concept generation.

Although we chose to keep the number of machine- and human-generated clusters the same in order to produce meaningful HM plots, our algorithm does allow varying the number of machine-generated clusters over a given set of data. We plan on developing guidelines for when reducing or increasing the number of machine-generated clusters would be of value. In testing our algorithm using a variable number of machine clusters, we found that fewer machine-generated clusters led to large bubbles and too large a number of clusters led to bubbles with a single concept. At least on our data set, humans seemed to do a reasonable job in setting the number of clusters relative to their concept pool. However, we predict there may be an advantage in increasing the number of machine-generated clusters when the average number of concepts per cluster is relatively high (say over 20% of the pool). As the goal of concept generation is to generate as many concepts as possible, we predict little advantage in reducing the number of clusters unless the number of concepts per cluster is very low (less than 2%). Note, none of the teams we evaluated were outside these ranges. Of course, any evaluation on the optimal number of clusters must be measured by the value to the design team processes.

Although not trivial, we will explore the possibility of using image extraction methods to extend the functionality of our method to use the sketch image in the half-sheet template in addition to the text in the concept description to add flexibility to our method, thus increasing its usefulness for design teams.

Our primary future research, however, will focus on interventions with product development teams that show patterns of over- and/or under-clustering in order to better understand how machine learning on clustering could support design teams to improve the

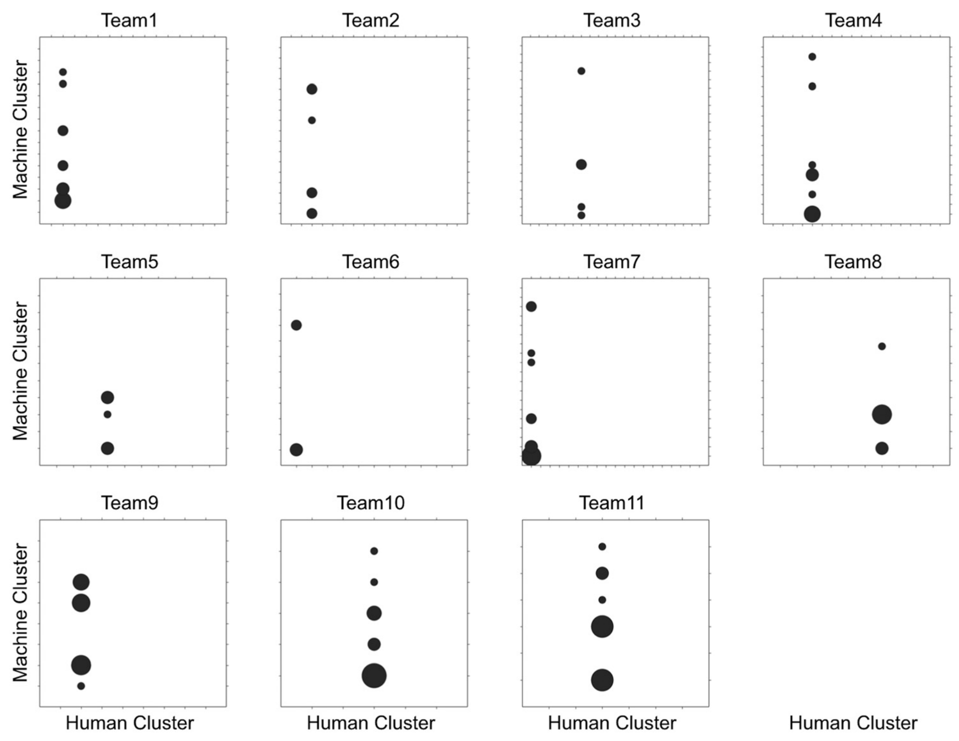


Fig. 8 Overview of under-clustering shown in teams' HM plots. Example: team 4 generated 12 concepts in cluster "children friendly designs" which the algorithm broke into six smaller clusters.

Table 6 Example of under-clustering pattern for team 4

Concept index	Concept name	Human label	Machine label
27	Children's energy playground	Children friendly designs	Energy; heat
29	Corn maze setup	Children friendly designs	Field; maze
25	Obstacle course	Children friendly designs	Obstacle course
26	Playground sailboat	Children friendly designs	Playground sailboat
21	Hot air balloon oscillator	Children friendly designs	Wind; device
22	Air dancer generator	Children friendly designs	Wind; device
23	Children spinning toy design	Children friendly designs	Wind; device
30	Fancy hat	Children friendly designs	Wind; device
31	Propeller cap	Children friendly designs	Wind; device
24	Swing set	Children friendly designs	Wind; generator
28	Teeter totter generator	Children friendly designs	Wind; generator
32	Oscillating wind sock	Children friendly designs	Wind; generator

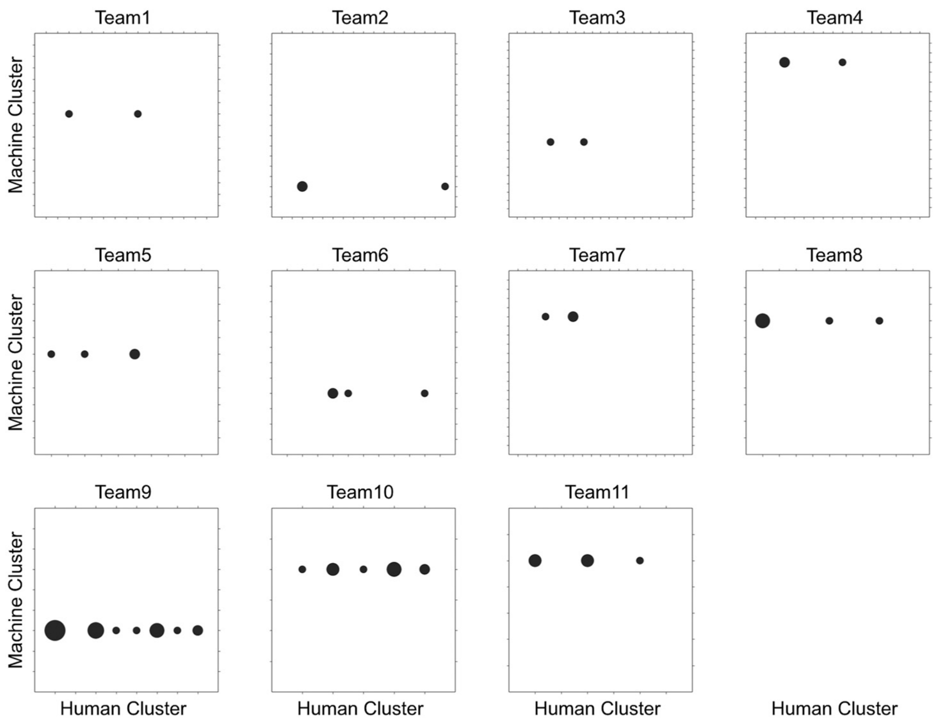


Fig. 9 Overview of over-clustering shown in teams' HM plots. Example: team 3 created two clusters "outside exercise" and "cycling" but the algorithm combined these into one larger cluster.

design process. Pilot tests with student teams in a Spring 2017 new product development class (ME 200)—similar to Fall 2016's ME 300—showed promising results in terms of student response to how they might use the HM plots. Two teams (out of 15 teams in the class) volunteered to participate in a pilot study. For example, one member of a team with a pattern of over-clustering recognized the challenges the team faced in clustering: "*During the concept generation stage, we had the crazy ideas we couldn't categorize which couldn't fit in the categories. Categories are automatically restrictive, so when we had crazy ideas, the categories really don't go together, so we ended up having big "others" (or miscellaneous things) which had everything that would fit. So I kind of like what has been done [in regards to the machine-generated clusters] to 'others'*."

A member of the other team that "under-clustered" commented: "In terms of concept generation, it might help toward the areas

Table 7 Example of over-clustering pattern for team 3

Concept index	Concept name	Description	Human label	Machine label
51	Hiking challenge app	An app that sets challenges for nearby hikes such as hike to top of grizzly peak in x amount of time use wearables to track	Outside exercise	App; challenge
56	Biking challenge app	Similar to hiking challenge app but focused on cyclists	Cycling	App; challenge

you might want to focus on, like the big bubbles, or other areas where there is nothing.… we can either pick out the best out of the list to pursue, or we can read through this and pick out the best part of each and come up with one better concept along with the concepts we already have."

Beyond the concept generation phase, improved clustering may also be useful in the convergent concept selection phase of product design to combine features across multiple concepts with similar functionality to get an improved design. Concept selection methods that build on concept clustering have only been lightly explored in the field of product development [15]. Thus, future plans for this work will focus on integrating our machine-learning model to consider both concept generation and selection processes together.

Acknowledgment

We thank the students and teaching staff in ME300 who participated in the concept clustering exercise. We also thank the student participants in ME 200 who participated in the concept clustering algorithm pilot tests.

Funding Data

- National Science Foundation, Division of Civil, Mechanical and Manufacturing Innovation (Grant No. 1334361).
- National Natural Science Foundation of China (Grant Nos. 51175287 and 51675299).

References

[1] Tan, P. N., Steinbach, M., and Kumar, V., 2005, "Data Mining Cluster Analysis: Basic Concepts and Algorithms," *Introduction to Data Mining*, Addison-Wesley, Boston, MA.
[2] Ulrich, K. T., and Eppinger, S. D., 2016, *Product Design and Development*, 6th ed., McGraw-Hill, New York.
[3] Goldstone, R. L., and Kersten, A., 2003, "Concepts and Categorization," *Handbook of Psychology*, Wiley, Hoboken, NJ.
[4] Smith, E. E., 1989, "Concepts and Induction," *Foundations of Cognitive Science*, M. I. Posner, ed., MIT Press, Cambridge, MA, pp. 501–526.
[5] Dong, A., and Agogino, A. M., 1996, "Text Analysis for Constructing Design Representations," *Artificial Intelligence in Design*, Springer, Dordrecht, The Netherlands, pp. 21–38.
[6] Wood, W. H., Yang, M. C., Cutkosky, M. R., and Agogino, A. M., 2014, "Design Information Retrieval: Improving Access to the Informal Side of Design," ASME J. Mech. Des., **136**(10), p. 101102.
[7] Salonen, M., and Perttula, M., 2005, "Utilization of Concept Selection Methods: A Survey of Finnish Industry," ASME Paper No. DETC2005-85047.
[8] Pugh, S., 1996, *Creative Innovative Products Using Total Design*, Addison-Wesley, Boston, MA, p. 544.
[9] Pugh, S., 1981, "Concept Selection: A Method That Works," International Conference on Engineering Design, Rome, Italy, Mar. 9–13, pp. 497–506.
[10] Roschuni, C., Goodman, E., and Agogino, A. M., 2013, "Communicating Actionable User Research for Human-Centered Design," Artif. Intell. Eng. Des. Anal. Manuf., **27**(2), pp. 143–154.
[11] Fisher, D. H., 1987, "Knowledge Acquisition Via Incremental Conceptual Clustering," Mach. Learn., **2**(2), pp. 139–172.
[12] Lloyd, S., 1982, "Least Squares Quantization in PCM," IEEE Trans. Inf. Theory, **28**(2), pp. 129–137.
[13] Shi, J., and Malik, J., 2000, "Normalized Cuts and Image Segmentation," IEEE Trans. Pattern Anal. Mach. Intell., **22**(8), pp. 888–905.
[14] Comaniciu, D., and Meer, P., 2002, "Mean Shift: A Robust Approach Toward Feature Space Analysis," IEEE Trans. Pattern Anal. Mach. Intell., **24**(5), pp. 603–619.
[15] Rokach, L., and Maimon, O., 2005, "Clustering Methods," *Data Mining and Knowledge Discovery Handbook*, Springer, New York, pp. 321–352.
[16] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C., 2003, "A Neural Probabilistic Language Model," J. Mach. Learn. Res., **3**, pp. 1137–1155.
[17] Mikolov, T., Chen, K., Corrado, G., and Dean, J., 2013, "Efficient Estimation of Word Representations in Vector Space," preprint arXiv:1301.3781.
[18] Levy, O., and Goldberg, Y., 2014, "Dependency-Based Word Embeddings," 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Baltimore, MD, June 23–25, pp. 302–308.
[19] Baroni, M., Dinu, G., and Kruszewski, G., 2014, "Don't Count, Predict! A Systematic Comparison of Context-Counting vs. Context-Predicting Semantic Vectors," 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Baltimore, MD, June 23–25, pp. 238–247.
[20] LeCun, Y., Bengio, Y., and Hinton, G., 2015, "Deep Learning," Nature, **521**(7553), pp. 436–444.
[21] Xu, W., Liu, X., and Gong, Y., 2003, "Document Clustering Based on Non-Negative Matrix Factorization," 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, ON, Canada, July 28–Aug. 1, pp. 267–273.
[22] Aggarwal, C. C., and Zhai, C., 2012, "A Survey of Text Clustering Algorithms," *Mining Text Data*, Springer, New York, pp. 77–128.
[23] Steinbach, M., Karypis, G., and Kumar, V., 2000, "A Comparison of Document Clustering Techniques," ACM Knowledge Discovery and Data Mining (KDD) Workshop on Text Mining, Boston, MA, Aug. 20–23, pp. 1–2.
[24] Zhao, Y., and Karypis, G., 2005, "Hierarchical Clustering Algorithms for Document Datasets," Data Min. Knowl. Discovery, **10**(2), pp. 141–168.
[25] Le, Q., and Mikolov, T., 2014, "Distributed Representations of Sentences and Documents," 31st International Conference on Machine Learning (ICML), Beijing, China, June 21–26, pp. 1188–1196.
[26] Han, E. H. S., and Karypis, G., 2000, "Centroid-Based Document Classification: Analysis and Experimental Results," European Conference on Principles of Data Mining and Knowledge Discovery, Lyon, France, Sept. 13–16, pp. 424–431.
[27] Manevitz, L. M., and Yousef, M., 2001, "One-Class SVMs for Document Classification," J. Mach. Learn. Res., **2**, pp. 139–154.
[28] Fuge, M., Peters, B., and Agogino, A., 2014, "Machine Learning Algorithms for Recommending Design Methods," ASME J. Mech. Des., **136**(10), p. 101103.
[29] Banerjee, S., Ramanathan, K., and Gupta, A., 2007, "Clustering Short Texts Using Wikipedia," 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23–27, pp. 787–788.
[30] Kuhn, A., Ducasse, S., and Gírba, T., 2007, "Semantic Clustering: Identifying Topics in Source Code," Inf. Software Technol., **49**(3), pp. 230–243.
[31] Fu, K., Chan, J., Cagan, J., Kotovsky, K., Schunn, C., and Wood, K., 2013, "The Meaning of 'Near' and 'Far': The Impact of Structuring Design Databases and the Effect of Distance of Analogy on Design Output," ASME J. Mech. Des., **135**(2), p. 021007.
[32] Fu, K., Cagan, J., and Kotovsky, K., 2010, "Design Team Convergence: The Influence of Example Solution Quality," ASME J. Mech. Des., **132**(11), p. 111005.
[33] Gorla, A., Tavecchia, I., Gross, F., and Zeller, A., 2014, "Checking App Behavior Against App Descriptions," 36th ACM International Conference on Software Engineering (ICSE), Hyderabad, India, May 31–June 7, pp. 1025–1035.
[34] Maalej, W., and Nabil, H., 2015, "Bug Report, Feature Request, or Simply Praise? On Automatically Classifying App Reviews," IEEE International on Requirements Engineering Conference (RE), Ottawa, ON, Canada, Aug. 24–28, pp. 116–125.
[35] Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., and Demirbas, M., 2010, "Short Text Classification in Twitter to Improve Information Filtering," 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, July 19–23, pp. 841–842.
[36] Dong, A., and Agogino, A., 1997, "Text Analysis for Constructing Design Representations," Artif. Intell. Eng., **11**(2), pp. 65–75.
[37] Pennington, J., Socher, R., and Manning, C. D., 2014, "Glove: Global Vectors for Word Representation," Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, Oct. 25–29, pp. 1532–1543.
[38] Wilkinson, L., and Friendly, M., 2009, "The History of the Cluster Heat Map," Am. Stat., **63**(2), pp. 179–184.