CIRP Annals - Manufacturing Technology 69(2020), to be published

CIRP Template

Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Design Transcription: Deep Learning based Design Feature Representation

Haluk Akay, Sang-Gook Kim (1)
Massachusetts Institute of Technology, Cambridge, MA, USA

The task of design feature transcription, or encoding the functional requirements and design parameters of a design, requires representing design data such that a machine can comprehend. Natural language processing, powered by deep neural networks trained on massive corpora of textual data, can map language into distributed vector representation space that machines can understand and retrieve. This work outlines how language models can be used to enhance early-stage design by separating the functional and physical domains, abstracting key functional requirements, and analysing systems to provide metrics for good design decision making, to facilitate a framework for hybrid intelligence.

Artificial Intelligence, Axiomatic Design, Deep Machine Learning

1. Introduction

A key part in any rigorous design process involves the steps of taking a list of system needs, translating these to a distilled set of key functional requirements, producing a design with physical parameters that satisfy each key requirement, and passing a form of analytic judgement to determine how “good” the design is, possibly in comparison to a baseline. Although this part of design is relatively well-defined, the complexity of these steps can increase greatly when the initial list of needs for a system is incomplete, contains extraneous information, or is otherwise noisy. Even given a successful distillation to key functional requirements (FRs), challenges remain. For the case of systems with multiple parent (higher level) FRs, each with child (lower level) FRs defining the detailed requirements of a system, understanding the effect of various design parameters (DPs) on each FR, and maintaining functional independence in a proposed concept becomes challenging as the system’s complexity increases. The concept of functional coupling is postulated in Axiomatic Design Theory as a metric for good design [1], with lack of functional independence being the cause of inferior designs or design failures [2,3].

Given the aforementioned challenges regarding distilling a set of FRs and matching DPs, and identifying functional coupling among them, the task of defining or describing them to be mutually exclusive and collectively inclusive is very difficult for even the most experienced human designers, resulting in many iterations of this process. Our proposed solution is the use of *hybrid intelligence* in which *human intelligence* co-evolves with *machine intelligence*. We are developing a deep learning-based design assistant which can extract succinct FRs and DPs from the design specifications and learn from past successes and failures to form a vast database to train machine intelligence for design itself [4].

Applying conventional Machine Learning (ML) methods to design and manufacturing is by no means a novelty; CIRP members have been using computational methods for decades, for example in the areas of process monitoring and parameter optimization [5], but quickly neural network computing became too intensive for hardware at the time, in the 1990s, to handle. The availability of cost-effective yet high performance Graphic Processing Units (GPUs) combined with massive amounts of image and textual data, as well as advances in the field of deep neural networks has resulted in powerful tools in Machine Learning, particularly in the domain of Natural Language Processing (NLP) and pattern recognition. Deep learning has allowed NLP to move away from a focus on syntax, grammar codes, and rule-based systems, and towards powerful models based on distributed vector representation methods. One of the primary tasks of deep learning-based NLP is representing language in a manner that machines can understand. Strings of alphabetic characters that make up language are quantitatively meaningless when presented, unprocessed, to a machine. However distributed vector language models [6-9] can convert words, sentences, and even entire documents into vectors in multidimensional space which can be understood by machines. Massive corpora of texts collected and curated from online archives of journalism and other readily available online knowledge bases are used to train these models. State-of-the-art models have demonstrated that geometric position of vector embeddings of language is a direct measure of their semantic meaning, learned from the context of the training dataset [10]. At a high level, these models at first naively convert language elements into a *one-hot-encoding* where each vector is simply the length of the number of words or sentences in the text, all full of 0s, except for a single 1 at the position of interest. Deep neural networks can then be trained on a variety of different prediction tasks using one of many different network architectures (both dependent on the model type), on a corpus of text which can include billions of words.

This paper proposes a process for facilitating *hybrid intelligence* by using deep language models to take an unstructured, “noisy” input of design statements from a novice designer, and output a design recommendation, according to the principles of Axiomatic Design, and powered by state of the art pre-trained language models [8,11,12]. In early-stage concept generation, a common challenge among novice designers is to remain “solution-neutral” and not leap to conclusions (DPs) before careful consideration of the functional domain (FRs). Therefore, when requesting a set of functional requirements from a novice designer, we can expect to be returned a mix of unstructured FRs and DPs. This work demonstrates how a classification model can be trained on synthetic, generatively produced examples of FRs and DPs to reliably identify and categorize unlabeled design statements passed as an input. Once a set of functions have been extracted from the user’s original input, the top, or “grandparent” FR(s) must be identified, in accordance with the top-down design thinking approach put forward by Axiomatic Design. This involves taking a paragraph-level sequence of unstructured FRs and abstracting the key FRs from this description of a design problem. This work demonstrates how a pre-trained abstractive text summarization model [12] can be applied to abstracting key FRs.

Finally, given a design defined by a set of FR and DP (what and how) pairs, a judgement must be made on how well the set of DPs satisfy the given FRs. Metrics of functional coupling, the bane of good system design practice, are defined in Axiomatic Design [13]. We have previously shown how semantic similarity can be used to approximate functional coupling [14]. This work further demonstrates how coupling can be quantified using deep language representations from pre-trained models such as Google's BERT [8].

2. Design Statement Classification

One of the most critical initial steps of the design process is distilling a design problem into a set of functional requirements (FRs). This is not a simple task for novice designers, who often find it challenging to remain solution-neutral at this stage and refrain from prematurely proposing design parameters (DPs). Therefore, there is a need to be able to reliably classify a sequence of design statements based on whether they are a true FR (what) or rather a DP (how). This problem falls into the domain of binary classification, visualized in the first step of Figure 1.

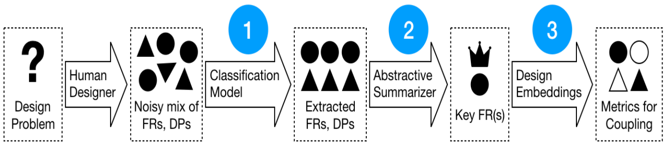

Figure 1. Process Flow for Design Transcription: determining metrics of functional independence given a design case

Binary classification is a well-defined Machine Learning task: a model can be trained on prior examples in order to make a determination of class, in this case FR or DP, for future unseen, unlabeled, examples. For this specific task, the key challenge was constructing a dataset of FRs and DPs, on the scale of approximately 10^5 labeled examples, to use for training and testing. Given the impracticality of manually creating, or scraping design documents for this magnitude of statements, a different approach was taken. A synthetic dataset of a balanced quantity of FRs and DPs was created using a pre-trained generative text prediction model, OpenAI's Generative Pre-trained Transformer GPT-2 [11] to auto-complete "seed phrases" which we provided. The design-oriented nature of the provided seed phrases guided the model to return a complete sentence, characteristic of an FR or DP depending on the seed.

Table 1. FR / DP Seed Syntactic Structure

	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5
FR Syntax	`[subject]`	`[imperative]`	`[verb]`	`[object]`	`[action]`
FR Example	*The system*	*needs to*	*allow*	*a user*	*to control*
DP Syntax	`[adjective]`	`[subject]`	`[modal]`	`[verb]`	`[action]`
DP Example	*One potential*	*solution*	*could*	*be applied*	*to support*

In order to supply approx. 10^5 seed phrases, a syntactic structure for the characteristic FR and DP was developed, as shown in Table 1. For each segment of each seed phrase, 9 different synonymous examples were collected manually. For example, the different forms Segment 1 of each FR seed could take included the list: [The design, device, product, system, structure, architecture, apparatus, appliance, contraption]. Given 9 examples and 5 segments, a total of $9^5 + 9^5$ or 118,098 unique FR and DP seed phrase combinations could be produced.

Next, each seed phrase was passed into a pre-trained GPT2 model, using the implementation released by the Hugging Face Transformer Library [15]. A maximum statement length of 25 words was set for each generation, and each of the 118,098 seed phrases were passed iteratively for auto-completion. Of these phrases, 91,259 were able to be generatively completed without issue. A random selection of these generated design statements are shown in Table 2.

Table 2. Selection of generated design statements

Design Statement (X) **Bold: seed phrase** *Italic: generatively completed statement*	Label (Y)
The product is obligated to help an individual to optimize *their own health and wellness.*	FR
The contraption is required to make it easy for somebody to ensure *that the device is not tampered with.*	FR
A supportive design could perhaps be made to adjust *the size of the body to accommodate the increased weight of the body.*	DP
An appropriate strategy may be modelled to support *the development of a new, more efficient, and more efficient energy source.*	DP

Each design statement was then embedded as a feature vector of 1024 dimensions in semantic space, using a pretrained model of BERT [8] implemented by the Hugging Face Transformer Library [15] to produce these representations. A balanced selection of 72,000 statement examples were used to train a Support Vector Machine (SVM), using the Scikit-Learn Library implementation [16], using a ν-value of 0.5, and the Radial Basis Function (RBF) kernel to lift the data into a separable feature space. For testing and evaluation purposes, 9,000 of the withheld statement examples (yet unseen by the trained SVM) were passed into the trained classification model, which misclassified only 8 examples, resulting in a test accuracy of 99.91%. Given the test data subset came from the same distribution as the training set, the high value of accuracy is expected. This trained model capable of identifying FRs from a mix of FRs and DPs allows the system to take in an unlabeled sequence of both statements and return just the functional requirements provided.

3. Abstraction of Key Functional Requirements

Once a design problem has been distilled to a set of its functional requirements (FRs) or rather the needs that a set of design parameters (DPs) must address, the top FRs must be abstracted so that the design can be developed with a top-down methodology. This task can be framed as taking a paragraph or document-level set of FRs and abstractively summarizing the text to return just the key FRs. Unlike binary classification, abstractive text summarization remains an ongoing challenge in the field of Natural Language Processing (NLP).

Table 3. Functional Abstraction using PreSumm-BERT [12]

Long-form user needs [17]	Abstracted Key FRs
The thermostat must be easy to install. This involves working with my existing heating or cooling system and being an easy project for a novice. The thermostat must be easily purchased. Another requirement is that the thermostat lasts a long time. It should be safe to bump into, resist dirt and dust, and have colors that don't fade over time, and when it's at its end of life, the thermostat should be recyclable. The controls of the thermostat must be precise. They must maintain temperature accurately within a small range of variability. It is important that the thermostat is easy to use. This means the user interaction is easy to understand and doesn't place significant demands on user memory. The thermostat should be able to be programmed from a comfortable position for the user, and it should be controlled remotely without requiring a special device. Also, it should work well right out of the box, with no set up needed. The thermostat should be easy to control manually, even for users with limited dexterity. The thermostat should be smart. It must adjust temperature during the day according to user preferences and adjust automatically during different seasons. The thermostat should be energy efficient and be a good investment and work normally even if electric power is suspended.	thermostat must be easy to install. the thermostat should be able to be programmed from a comfortable position for the user. it should be easier to control manually, even for users with limited dexterity

Using deep neural networks, this task can be modeled as a sequence-to-sequence problem. If the vectorized inputs are considered to be the sequence $x = [x_1 \dots x_m]$ then the target summary can be determined based on the conditional probability $p(y_1 \dots y_n \mid x_1 \dots x_m)$, resulting in the summary $y = [y_1 \dots y_n]$ where n << m. To generate the sequences x and y, an encoder and decoder are required respectively. Previously referenced models for encoding text to vectors exist [6-9]. Yang proposes using BERT as an encoding model, and a 6-layer Transformer neural network [12], trained on labelled datasets of news articles and corresponding summaries, as the decoder. Using the Hugging Face Library implementation of the encoder-decoder model [15], we can demonstrate how the top FRs can be abstracted from a longer sequence. Table 3 demonstrates how the decomposed user needs for a smart thermostat [17] can be functionally abstracted to a few key FRs. In the abstracted summary, less important FRs such as "colors that don't fade over time" are left out, and the key FRs focus on easy usability and control. Using this pre-trained abstractive text summarization model, key FRs can be functionally abstracted from a sequence of unstructured FRs, to find the root requirements of a design.

4. Quantifying Functional Independence in Systems Design

As introduced previously, lack of functional independence (coupling) is the bane of good systems design. When considering a system with *n*-number of FRs and associated DPs addressing each requirement, Axiomatic Design thinking uses the "Design Matrix" [1] of equation 1 to describe the relative effect of each DP on each FR in the system.

$$\begin{Bmatrix} FR_1 \\ \vdots \\ FR_n \end{Bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix} \begin{Bmatrix} DP_1 \\ \vdots \\ DP_n \end{Bmatrix} \tag{1}$$

For the case of an uncoupled design, the design matrix [A] would be diagonal, meaning $A_{ij} = 0$ when $i \neq j$. In this case, each DP satisfies a single FR in an ideal design. For the case of coupled design, any change in an FR cannot be addressed by an adjustment of any combination of DPs without also affecting other FRs. Simple designs may be classified as coupled, uncoupled, or de-coupled (as is the case when the design matrix is triangular). However, in more complex cases, designs may not be easily categorized into these discrete buckets. For such cases, metrics of functional independence have been developed to quantify coupling on a continuum [13,18].

Two metrics have been identified for functional independence. The first is *Reangularity* (R), which reflects the degree to which different DPs have the same effect on the set of FRs [13], a measure of orthogonality between DPs. R is related to the cosine similarity of the elements of the design matrix from equation 1 relating each DP component to the FR vector and can be generalize for an n-dimensional case as follows in equation 2 [13].

$$R = \prod_{\substack{i=1,n-1 \\ j=1+i,n}} \left(1 - \frac{\left(\sum_{k=1}^{n} A_{ki} A_{kj}\right)^2}{\left(\sum_{k=1}^{n} {A_{ki}}^2\right)\left(\sum_{k=1}^{n} {A_{kj}}^2\right)}\right)^{1/2} \tag{2}$$

DPs may be independent of each other yet may still affect multiple FRs . The next metric, *Semiangularity* (S) is a measure of the degree to which each DP affects one and only one FR of the design. S can be expressed as the product of the absolute values of the diagonal elements of the design matrix, normalized in equation 3 [13].

$$S = \prod_{j=1}^{n} \left(\frac{|A_{jj}|}{\left(\sum_{k=1}^{n} A_{kj}^2\right)^{1/2}} \right) \tag{3}$$

R and S values close to 0 indicate the worst case (fully coupled design), while values close to 1 indicate the ideal, perfectly functionally independent case. While this framework is powerful for determining relative degrees of coupling, it has been difficult to implement for real-world application as accurately assigning continuous values to the design matrix *A* is challenging.

While models such as BERT have demonstrated state of the art performance for general language tasks and have been fine-tuned for application to specific domains such as producing clinical embeddings [21], they have mixed results when it comes to direct application for producing representations for design analysis. It has been shown that for simple design cases, functional independence can be computed using BERT embeddings of design features expressed in natural language [14]. A demonstration of how semantic similarity learned by deep language representation models such as BERT translates to the functional domain is illustrated using the examples of faucet and refrigerator design, classic cases in Axiomatic Design of the concept of functional independence.

Faucet A Faucet B Refrig. A Refrig. B. Refrig. C

Figure 2. From left to right: Faucet designs A(coupled) and B(uncoupled). Refrigerator designs A and B (coupled) and modular refrigerator C (uncoupled) [22][23]

Considering first faucet design, the two key FRs can be understood as allowing the user to control temperature and flow rate of water. Although both faucets A and B in Figure 1 allow control of these parameters, faucet A is coupled because the two valves affect both FRs when adjusted, leading to a complex fine-tuning process if the user wants to reach a specific desired temperature and flow rate. Faucet B, however, has two levers controlled by adjusting the handle vertically or horizontally, such that flow rate and water temperature can be independently controlled, and is therefore an example of an uncoupled design.

Table 4 Descriptions detailing Faucet and Refrigerator designed used to compute metrics illustrated in Figure 2

Faucet FRs	***Allow control of water temperature***	***Allow control of water flow rate***
Faucet A DPs	*One valve to control flow rate of cold water*	*One valve to control flow rate of hot water*
Faucet B DPs	*One lever to control temperature of water*	*One lever to control flow rate of water*

Refrig. FRs	***Provide easy access to items stored inside***	***Minimize energy loss when door is opened***
Refrig. A DPs	*Side wall hinged to swing outwards and uncover an entire side panel of refrigerator enclosure. While the door is open, cold air is easily exchanged with room-temperature air through the side of the refrigerator and increases energy loss during access.*	*Doors and walls of refrigerator enclosure made from materials that minimize heat transfer.*
Refrig. B DPs	*Top face slides away to uncover the contents stored in the refrigerator for access from above. It is difficult to access items that are not close to the top or if the refrigerator is full.*	*Doors and walls of refrigerator enclosure made from materials that minimize heat transfer.*
Refrig. C DPs	*Items stored in compartments, accessed by hinged doors or drawers. Items are easily accessed, and cold air is only lost from a small volume.*	*Doors and walls of refrigerator enclosure made from materials that minimize heat transfer.*

In refrigerator design, two key FRs can be stated as allowing the user ease of access to items within and minimizing energy loss during access. Design A's side doors allow easy access to items stored inside the refrigerator, but also allow cool air to fall out every time they are opened. Design B's top door keeps cool air inside but also does not allow easy access to items at the bottom, and so is also a coupled design. Design C's uncoupled modular design [22] minimizes energy loss by compartmentalization while preserving ease of access. Originally designed for shared kitchen spaces by providing each user a small personal space, the design is also energy efficient during use.

The coupling metrics in Figure 3 were computed by having an expert human designer craft detailed but succinct descriptions of the FRs and DPs comprising each design case, as shown in Table 4. Using these sentence embeddings were produced using a pre-trained model of BERT. Finally, the design matrix A was computed, and the values for Reangularity and Semiangularity were solved.

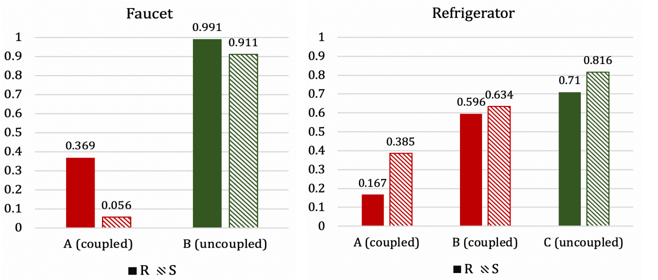

Figure 3. Functional independence is reflected in BERT embeddings of detailed design descriptions used to compute R and S values for faucet (left) [14] and refrigerator (right) designs.

For these simple design cases, it is clear that there is a link between semantic similarity and the functional domain. Intuitively, it makes sense that the DP description of a 'valve to control the flow rate of hot water' will be semantically similar to both FRs, being flow rate and water temperature, while the DP description of a 'lever to control temperature of water' shares similarity with just the temperature FR, with no mention of flow rate.

A key limitation of this method for measuring coupling in a design is the sensitivity of the sentence embedding model to the description of design features, written in natural language by an experienced designer in the cases present previously. Small changes in phrasing which do not significantly alter the key functional description of the FR or DP will result in different sentence vectors produced by the language model. Depending on the word choice and phrasing of the design statements, the measurements for R and S will be affected.

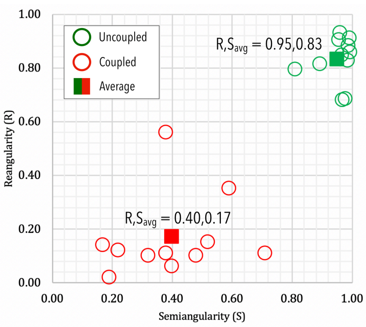

Figure 4. Sensitivity to word choice, faucet design case

To gauge the sensitivity of the measurements to word choice, the descriptions of the faucet design were perturbed by making changes in word choice and inserting extra phrases, to make 10 variants each on the original descriptions of the coupled and uncoupled cases. From the original description in Table 4, the words [control, water, lever, water, allow, valve] were substituted with 2 different synonyms each, and the phrases [the user to, someone to] were inserted into the FRs before recalculating R, S, values, which are shown in Figure 4. For minor re-wording and re-phrasing, each description is observed to cluster in a position indicative of the design's functional independence.

6. Conclusion

At the early stages of the design process, designers can benefit from hybrid intelligence by collaborating with machines to augment important decision making. Deep learning models in the area of natural language processing provide an opportunity for representing designs in a way that machines can understand, to facilitate hybrid intelligence at this stage in design. It was demonstrated how text generation models can be used to create massive labeled datasets of design statements to train classification models on separating problems (FRs) from solutions (DPs). Abstractive text summarization encoder-decoder models can be used to abstract key FRs, which can then be quantified by deep language representation models to compute metrics for functional independence for the machine to provide advice for good design decision making. For simple, well defined design cases, this was demonstrated to provide meaningful results reflecting the true coupled or uncoupled nature of the designs.

These models can be sensitive to changes in wording of the input language such that a system for filtering out noisy and redundant design descriptions to succinct and meaningful ones is needed for a reliable design transcription process. If these models can be fine-tuned with more generatively created design domain datasets, then they can be trained to filter noisy and unstructured design descriptions to succinct and minimally transcribed design features for effective and precise vectorization of them. This is similar to the power of deep learning which has been very successful in pattern recognition and speech recognition.

Acknowledgements

This work was supported by SUTD IDC Flagship Project Funding, MIT/SenseTime Alliance on Artificial Intelligence, and the National Science Foundation, Leading Engineering for America's Prosperity, Health, and Infrastructure (LEAP HI) program, award number 1854833.

References

[1] Suh, N. P. (1990). *The principles of design* (No. 6). Oxford University Press.
[2] Suh, N. P., Kim, S. H., Bell, A. C., Wilson, D. R., Cook, N. H., Lapidot, N., & von Turkovich, B. (1978). Optimization of manufacturing systems through axiomatics. *Annals of the CIRP*, *27*(1), 383-388.
[3] Suh, N. P. (1995). Axiomatic design of mechanical systems.
[4] Kim, S. G., Yoon, S. M., Yang, M., Choi, J., Akay, H., & Burnell, E. (2019). AI for design: Virtual design assistant. *CIRP Annals*, *68*(1), 141-144.
[5] Choi, G. H., Lee, I. K., Chang, N., & Kim, S. G. (1994). Optimization of process parameters of injection molding with neural network application in a process simulation environment. *CIRP annals*, *43*(1), 449-452.
[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
[7] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
[8] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
[9] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5754-5764).
[10] Goldberg, Y., & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.
[11] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.
[12] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.
[13] Rinderle, J. R. (1982). *Measures of functional coupling in design* (Doctoral dissertation, Massachusetts Institute of Technology).
[14] Akay, H., Kim S-G. (2020). Measuring functional independence in design with deep-learning language representation models. Procedia CIRP (accepted for publication).
[15] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Brew, J. (2019). Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*.
[16] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.
[17] Ulrich, K. T. (2003). *Product design and development*. Tata McGraw-Hill Education.
[18] Suh, N. P., & Rinderle, J. R. (1982). Qualitative and quantitative use of design and manufacturing axioms. *Cirp Annals*, *31*(1), 333-338.
[19] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).
[20] Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., ... & Hon, H. W. (2019). Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems* (pp. 13042-13054).
[21] Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jin, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323*.
[22] Buchberger, Stefan (2008). Modular Fridge Design, finalist, Electrolux Design Lab 2008 Competition, University of Applied Arts, Vienna
[23] Photography by Rebecca Wiggins, Sasikan Ulevik, and Phillip Pessar from unsplash.com
[24] Buitinck, Lars, et al. "API design for machine learning software: experiences from the scikit-learn project." *arXiv preprint arXiv:1309.0238* (2013).