Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 91 (2020) 528–533

www.elsevier.com/locate/procedia

30th CIRP Design 2020 (CIRP Design 2020)

Measuring functional independence in design with deep-learning language representation models

Haluk Akay, Sang-Gook Kim*

Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

* Corresponding author. Tel.:617-452-2472,. *E-mail address:* sangkim@mit.edu

Abstract

Measuring functional coupling in complex systems is an important task for good design practice, though historically it has been an art of subjective judgement. With the recent advancements in Deep Learning and Natural Language Processing, functional requirements (FRs) and design parameters (DPs), which are expressed as words and sentences, can be represented in a vector space. The sentence embedding model, BERT, was used in this paper to vectorize FRs and DPs, to calculate functional independence and to study how metrics for functional coupling measurement can be enhanced. It was found that semantic similarity among FRs and DPs, represented in vector space, could be used to compute quantitative values for metrics of functional independence. It was also found that design cases where coupling was unambiguous yielded the best results, while cases where laws of physics needed to define the FR-DP relationship did not transliterate well to the natural language used to express the FR-DP highlighted the limitations of the model in its current state. This study, however, demonstrates a great opportunity to develop a robust, fine-tuned design language representation model for accurately measuring functional independence as a part of our effort to enhance design intelligence.

Keywords: Axiomatic design, functional independence, deep learning, language representation

1. Introduction

The task of large and complex systems design relies heavily on expert's heuristic knowledge and insight in delivering a systems solution. Success at this scale results in elegant and efficient engineering, social or service systems that can provide users and the society with a better quality of life, while failure can result in long-delayed, over-budgeted, and sometimes unfinished projects and even catastrophic loss of life. A well-established and widely-accepted metric for good design is the concept of functional independence [1], where maintaining functional independence is key for good design, and functional coupling can lead to inefficiency and aforementioned catastrophic failure. Axiomatic Design (AD) has provided a way of design thinking that the heuristic-based system design could be better structured with the concept of domains and securing functional independence (Axiom One). But many still find it difficult to apply AD principles, such as functional independence, to practical problems since assessing functional coupling requires subjective judgement and substantial amount of experience. While existing designs can be analyzed for functional coupling retrospectively by an experienced systems designer, it is difficult to measure coupling during the early design process prospectively, especially when the design parameters and functional requirements are difficult to quantify and normalize. For this reason, most efforts to apply AD principles to industrial practice have fallen short of becoming widely applied tools for design success yet.

A framework of AI for design was developed by the authors to facilitate the functional thinking of junior designers by

10.1016/j.procir.2020.02.210

representing designer's intention in the syntax of functional requirements, assisting human designers in making good design decisions with AD principles [2]. Deep learning-based algorithms have been applied to translate (encode) user needs and specifications to a collection of functional requirements (FRs) that can be vectorized and then understood by the machine learning tools to create structure and hierarchy of them. Once vectorized, FRs and design parameters (DPs) could utilize the rich set of AD principles, theorems and corollaries to assess the designer's decision and provide adequate advices. This paper reports on whether vectorized FRs and DPs could be assessed algorithmically whether they are coupled or not.

Recent breakthroughs in the field of machine learning as a result of relatively cheap graphic processing units (GPUs) that can power deep neural networks, combined with large amounts of readily available data have resulted in a resurgence in interest in artificial intelligence (AI). In particular, the field of natural language processing (NLP) has seen great advancements as industry seeks to make use of massive amounts of text created by queries on search engines, posts on social media, and even published academic papers. While unprocessed words and statements by themselves have no quantitative value for computing similarity, sentiment, or other attributes of natural language, powerful language models trained on large corpora of text can represent words and sentences in multidimensional vector space. Work in the field of applying neural networks to statistical language modeling can be traced to Rumelhart, Hinton et. al. [3], but with the aforementioned availability of corpuses containing billions of words, recent word embedding techniques, specifically the work of Mikolov et. al [4] has resulted in models that can vectorize text such that simple linear translations of word vectors yield intuitive responses. As demonstrated in [5], the sum of vectors for `Germany` and `capital` when added together is closest to the vector for `Berlin`, showing how linguistic patterns can be accurately captured by deep learning-based language representation models. Further NLP breakthroughs in vectorizing phrases, whole sentences, and even documents have followed. Most recently, a new language representation model, "Bidirectional Encoder Representations from Transformers" or BERT [6], has released a pre-trained language model demonstrating record-breaking performance on eleven benchmark NLP tasks. The field of NLP is so fast-paced that during the work on this paper itself, BERT was outperformed by a new autoregressive model named "XLNet" [7], showing new state-of-the-art tools emerging from the research area of natural language representation.

This work leverages deep language representation models such as BERT to quantify FR-DP coupling relationships and to facilitate the integration of AI and human intelligence in the domain of design, building on the concept of hybrid intelligence as introduced in Kim et. al [2].

2. Axiomatic Design & Deep Language Representation

Among a multitude of theories for how to best execute good design, Axiomatic Design (AD) stands out as a principle-based methodology to designing systems [8]. AD provides a framework for mapping between the functional domain and physical domain. While it is a powerful tool for facilitating early-stage top-down systems design thinking in research and academic settings, it has not been widely incorporated into industrial practice. AD holds two key Axioms: that functional independence must be maintained, and complexity must be minimized. Functional independence relates to the relationships between "what" and "how" of a design. Functional requirements (FRs) represent the former, and are derived from users' needs: the problem the design addresses. Design Parameters (DPs) are "how" a solution solves the problem. The relationship between FRs and DPs can be mathematically represented using equation 1, known as the "Design Equation" [1]:

$$\{FR\} = [A]\{DP\} \tag{1}$$

{FR} represents the functional requirement vector (design goals) and {DP} is the design parameter vector (how these goals will be addressed). The remaining term [A] is the design matrix, each element of which represents a relation between a component of the FR vector to a component of the DP vector. For example, a design with n FRs and m DPs could be represented by equation 2:

$$\begin{Bmatrix} FR_1 \\ \vdots \\ FR_n \end{Bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nm} \end{bmatrix} \begin{Bmatrix} DP_1 \\ \vdots \\ DP_m \end{Bmatrix} \tag{2}$$

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. The significance of functional coupling in design, especially of complex systems, cannot be understated. Small design updates to one portion of a system can have undesired and even initially unnoticed effects on other aspects of the overall design in the case of a coupled design. A classic comparison of a coupled and uncoupled design in everyday life is apparent in faucet design. Considering the FRs of a faucet to allow the user to control temperature of water and control flow rate of water, a commonly implemented but coupled instance of faucet design has two valves (Figure 1a); one for cold and one for hot water. Controlling temperature and flow rate is difficult for the user because of functional coupling; adjusting each knob affects both variables. The uncoupled instance of the design (Figure 1b) allows the user to adjust flow rate with a vertical lever, and temperature by moving the lever horizontally.

Fig. 1. (a) right: coupled faucet; (b) left: uncoupled faucet design. Photography by Rebecca Wiggins and Sasikan Ulevik on unsplash.com

While clear in some simple cases, functional coupling can be difficult to measure in more complex designs with multiple FRs and DPs, or where the relationship between the functional and physical domains can be ambiguous. Efforts have been made to determine metrics to measure degree of coupling in such cases, based on the representation of FRs and DPs as vectors without actually quantifying them [9-10]. FRs, by definition, are independent, and so can be represented as orthogonal vectors. DP isograms, when plotted in the functional domain, may or may not be parallel to FR axes, depending on how many FRs they affect. For a design with two FRs and DPs, the design can be visualized with isograms as in Figure 2 below.

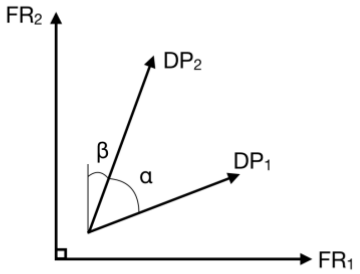

Fig. 2. DP isograms plotted in the functional domain.

There are two metrics for coupling, Reangularity (R) and Semiangularity (S). Reangularity R reflects the degree to which different DPs have the same effect on the set of FRs [9], essentially a measure of orthogonality between DPs, corresponding to the angle α between DPs in Figure 2. R is related to the cosine similarity of the elements of the design matrix relating each DP component to the FR vector, and can be generalize for an n-dimensional case as follows in equation 3 [9].

$$R = \prod_{\substack{i=1,n-1 \\ j=1+i,n}} \left(1 - \frac{\left(\sum_{k=1}^{n} A_{ki}A_{kj}\right)^2}{\left(\sum_{k=1}^{n} {A_{ki}}^2\right)\left(\sum_{k=1}^{n} {A_{kj}}^2\right)}\right)^{1/2} \tag{3}$$

The second metric, Semiangularity S reflects the degree to which each DP affects one and only one FR of the design [9]. DPs may be independent of each other yet may still affect multiple FRs as can be visualized in the case of an orthogonal pair of DPs oriented at a non-parallel angle to the functional domain. S can be expressed as the product of the absolute values of the diagonal elements of the design matrix [A], normalized as in equation 4 [9].

$$S = \prod_{j=1}^{n} \left(\frac{|A_{jj}|}{\left(\sum_{k=1}^{n} A_{kj}^2\right)^{1/2}} \right) \tag{4}$$

When Reangularity and Semiangularity of a design are close to zero, this indicates the design is fully coupled (worst case); when R and S values are close to 1, this is an indication of functional independence, an ideal design case.

This framework for quantifying coupling in design [9] was interesting but has been difficult to apply to real-world design cases because of the challenge presented by determining the values of the elements in the design matrix [A] accurately, when various FR-DP pairs may deal with quantities measured in different units, or when the FR in question is simply qualitative at the early stage of design. It is at this limitation of measuring functional independence with AD principles that deep language representation models become highly applicable.

3. Language and function embedding

Words and phrases, expressed as strings in digital texts, have no quantifiable meaning in their raw form. Mathematically there is no way to quantify the relationship between the word "temperature" and "flow" simply based on their form as a sequence of letters from a given alphabet. The simplest and least efficient method of converting all the words in the dictionary to multi-dimensional vector space (where mathematic manipulation can occur) would be to create an n-dimensional vector for each word where n is the number of total words in the dictionary. Using the "one-hot encoding procedure," each word vector would be made up entirely of zeros except for the position that the word in question occupies alphabetically.

The goal of vectorizing words and sentences is to place them in n-dimensional space such that language of similar context occupies nearby positions. State of the art sentence embedding models such as BERT [6] use deep neural networks to accomplish this task. In the BERTLARGE model configuration, a neural network with 24 layers is pre-trained on unlabeled data from the BooksCorpus and the English Wikipedia, amounting to more than 3 billion words in the form of documents, because continuous sequences of sentences are important for the model to learn context. The same model architecture is then fine-tuned by initializing the same neural network with parameters obtained from pre-training and learning this time with labeled data. The network architecture used by the developers of BERT is a multi-layer bidirectional transformer.

Based on the transformer architecture proposed by Vaswani et. al [11], BERT is capable of "reading" text bi-directionally, where word tokens can consider context to both right and left in a sentence. The model is trained on two tasks. Given an input of a small number of sentences, a random selection of tokenized words is masked, and the model trains by predicting correct words by choosing from a probability distribution of appropriate words, a process known as a Masked Language Model (MLM). The second task is simply predicting a subsequent sentence, which helps the model gain a sense of context. At the time of its release, BERT reached state-of-the-art results in numerous benchmarks for language embedding performance. For the experiments demonstrated in this paper, the pretrained BERTBASE model (12-layer architecture) was used to demonstrate how powerful state-of-the-art word embedding tools integrated with theories of functional independence can be for measuring coupling in design.

Both semantic and sentiment understanding is a key goal of language representation models such as word2vec and BERT developed by social media companies (Facebook and Google respectively). For example, a task of interest in this industry is differentiating between sarcastic statements and text suggesting the author wishes to seriously cause harm. This paper proposes that the same tool can be used for design analysis in the functional domain by representing functional requirements and design parameters in vector space.

4. Measuring Functional Independence

Given the framework that Axiomatic Design provides for translating FRs and DPs into linear algebra equations, and the properties of the design matrix that relate to measures for coupling in terms of Reangularity and Semiangularity, language representation in the form of vectorized phrases proves to be a valuable tool for assessing functional coupling quantitatively. Rather than considering FRs and DPs as design variables measured with units in disagreement such as the faucet example where the units for flow rate [volume / time] and temperature [degrees] are incongruous, we can instead consider FRs and DPs as design statements positioned in the same multidimensional vector space. In this space, contextual relationships between phrases can lend an understanding of coupling if manipulated correctly.

If we recall equation (2) which describes how multiple FR and DP pairs are related via the design matrix [A], we can consider a somewhat idealized case where the number of FRs is equal to the number of DPs. This case is not necessarily coupled or uncoupled. Leveraging any number of pre-trained language representation models, each design statement (FR + DP chosen) can be converted from a string of characters to an n dimensional vector. Later, for demonstration purposes, BERTBASE will be used for this task. By inserting the statement vectors into a design equation with m FR-DP pairs, we end up with the following expression for equation 5.

$$\begin{Bmatrix} [FR_{1_1}, FR_{1_2} \cdots FR_{1_n}] \\ \vdots \\ [FR_{m_1}, FR_{m_2} \cdots FR_{m_n}] \end{Bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mm} \end{bmatrix} \begin{Bmatrix} [DP_{1_1}, DP_{1_2} \cdots DP_{1_n}] \\ \vdots \\ [DP_{m_1}, DP_{m_2} \cdots DP_{m_n}] \end{Bmatrix} \quad (5)$$

Now both the {FR} and {DP} matrices have dimensions m by n where m represents the number of FR-DP pairs, and n represents the dimensionality of the statement vectors. The design matrix [A] still has dimensions m by m, and remains the most interesting component of the expression. If we recall the expressions for Reangularity R and Semiangularity S from equations 3 and 4, we see that using the values of elements from [A], measures of functional coupling can be determined from the vectorized design statements of equation 5. The matrix A can be solved for using regression, and the value of R and S, each on a scale of 0 to 1 can be computed. R values close to zero indicate great similarity between DPs, while values closer to 1 indicate orthogonality between DPs. However, even orthogonal DPs can be coupled if not aligned with FRs in the functional domain. S values close to 1 indicate perfect alignment of FR-DP pairs, such that a design with perfect functional independence would have an R and S value both of 1. The opportunities and limitations of directly inserting statement vectors into the design equation are demonstrated in the subsequent section.

5. Experiments

In this section, four designs of two separate products are considered. For each product, an example of a coupled design followed by a design where functional independence is maintained are compared, and metrics for coupling are computed using equation 5 to evaluate the design matrix, and equations 3 and 4 to evaluate Reangularity *R* and Semiangularity *S*. Each design is described by two FR-DP pairs for simplicity. The design statements are converted to n-dimensional vectors where n = 768 using a BERTBASE pre-trained model. The Hugging Face implementation [12] of Devlin et. al's BERT model was used to produce sentence embeddings for the following examples

5.1. Water Faucet

Consider the two common designs for a faucet, illustrated previously in figure 1. The first design featuring separate valves for hot and cold water flow is clearly coupled because adjusting either knob affects both flow rate and temperature, making it difficult for the user to control both at once. The FRs and DPs for a fully coupled case can be expressed with matrices in equation 6, and in natural language below.

$$\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} = \begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \end{Bmatrix} \quad (6)$$

Faucet (coupled):
FR1: *Allow control of water temperature*
FR2: *Allow control of water flow rate*
DP1: *One valve to control flow rate of cold water*
DP2: *One valve to control flow rate of hot water*

For the coupled case, R is 0.369, and S is 0.056. These metrics suggest functional coupling in this design. Next, we consider the second design illustrated in figure 1b, which features a lever that when moved horizontally controls water temperature, and when adjusted vertically increases flow rate. This is a functionally independent case, described by equation 7, as the user can independently control temperature and flow rate. The natural language description of the design is below.

$$\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} = \begin{bmatrix} \times & 0 \\ 0 & \times \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \end{Bmatrix} \quad (7)$$

Faucet (uncoupled):
FR1: *Allow control of water temperature*
FR2: *Allow control of water flow rate*
DP1: *One lever to control temperature of water*
DP2: *One lever to control flow rate of water*

For this uncoupled case, R is 0.991 and S is 0.911. These metrics reflect a nearly perfectly functionally independent design. The isograms for each faucet design case can be used to illustrate, in Figure 3, how the computed values of R and S reflect functional coupling.

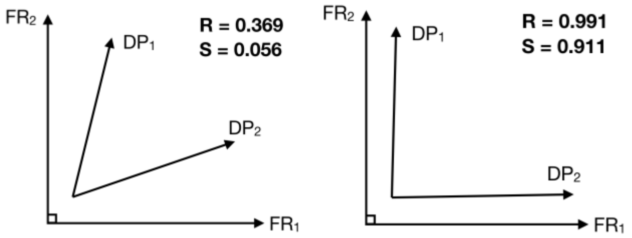

Fig. 3. Functional Domain Isograms for Coupled (left) and Uncoupled (right) faucet design cases

5.2. Steam Engine

Next, we consider steam engine design from the 18th century. For more than half a century, the "Atmospheric" steam engine designed by Thomas Newcomen in 1712 was the point of reference for industry at the time, used extensively for a number of applications for decades [13]. The engine cycle operated by pulling a piston head upwards in a cylinder using some application-related weight fixed to a pulley. During this upstroke, a boiler injects steam into the piston-cylinder assembly. To pull the piston head down again and raise the weight at the end of the pulley, the cylinder is cooled to below atmospheric pressure (hence the name) by spraying cold water into the cylinder. This creates a partial vacuum which pulls the piston back down to the bottom of the cylinder.
This is a case of coupled design because cooling and reheating occurs in the same cylinder, increasing the cycle time needed to create a partial vacuum before the next stroke. The design can be described in natural language, decomposing the DP into two DPs to address each FR.

Newcomen Steam Engine (coupled):
FR1*: Lower pressure to a partial vacuum in the cylinder*
FR2: *Raise pressure back to atmosphere in the cylinder*
DP1: *Condense steam by cooling the cylinder with cold water*
DP2: *Draw high temperature steam from the boiler into the cylinder*

For the coupled case, *R* is 0.004, and *S* is 0.596. The especially low value for Reangularity suggests that the two DPs are very similar and not orthogonal. Such a design where two FRs are satisfied by one DP is inherently coupled and can be described by equation 8.

$$\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} = \begin{bmatrix} \times \\ \times \end{bmatrix} \{DP_1\} \tag{8}$$

We can visualize these as isograms in the functional domain, where they reflect the fact that the Newcomen engine uses one DP to address two FRs, resulting in a coupled design, shown in Figure 4. It was more than fifty years before James Watt, while repairing a Newcomen engine, identified the inefficiencies of its design and decoupled functionality by introducing a new component: a condensation chamber which could be cooled separately to create a vacuum to pull the piston down to the bottom of the cylinder [13]. The James Watt steam engine design can be described below in natural language, and is also represented by equation 7.

Steam Engine (uncoupled):
FR1*: Lower pressure to a partial vacuum in the cylinder*
FR2: *Raise pressure back to atmosphere in the cylinder*
DP1: *Draw steam into a separate low temperature condensation chamber connected to the cylinder by a valve*
DP2: *Draw high temperature steam from the boiler into the cylinder*

For this uncoupled case, *R* is 0.437 and *S* is 0.892. The addition of an independent design parameter (the condensation chamber) is reflected in the increase in value of Reangularity.

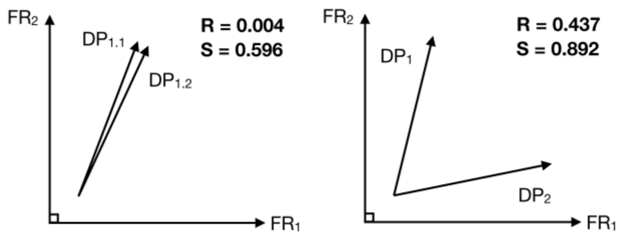

Fig. 4. Functional Domain Isograms for Single DP (coupled) (left) and Uncoupled (right) steam engine design cases

The results of these demonstrations are summarized in Table 1.

Table 1. Results for Reangularity *R* and Semiangularity *S*.

Design Case	*Reangularity (R)*	*Semiangularity (S)*
Faucet (coupled)	0.369	0.056
Faucet (uncoupled)	0.991	0.911
Steam engine (coupled)	0.004	0.596
Steam engine (uncoupled)	0.437	0.892

6. Discussion

Two unique design areas were used to demonstrate the applicability of language representation models such as BERT in measuring functional independence. The faucet design case yielded the most difference between metrics for coupled and uncoupled designs. When statements are converted to vectors, similar sentences and phrases occupy a closer location in vector space than unrelated phrases. For the faucet design, it was very simply stated in a succinct phrase in the uncoupled case that one DP clearly addressed temperature, and the other addressed waterflow. The control of these two exact variables being the top-level FRs meant high vector similarity, reflected in the computed values for Reangularity and Semiangularity.

The single-DP coupled case of the Newcomen steam engine showed how such designs can also be identified in vector space. The solution to address each FR both resulted in raising/lowering pressure in the same cylinder, leading to high sentence similarity between DPs, reflected in the functional domain where indeed one DP was being tasked to both FRs. Reangularity of near zero reflects a design where the DPs are nearly identical.

Design cases with no more than two FR – DP pairs were demonstrated, with very precisely and concisely stated descriptions. When integrated into a user-facing system, more "noisy" descriptions can be expected from novice designers attempting to make better design choices with this computational aid. In the model's current form, if the level of "noise" in these descriptions is increased to the point where they have inconsistencies, or convey only partially complete information, then the feature representation model will less consistently produce a vector indicating the key semantic context of the sentence, depending on how poorly stated is the input. For these cases, a pre-processing step would be needed before the representation model could produce meaningful vectors that accurately represent the semantics of each design statement. Longer descriptions, possibly including multiple sentences, with verbose descriptions, can be abstracted to key FRs and DPs to provide concise and precise inputs to the feature representation model.

7. Conclusion

The concept of functional independence in the case of system design was introduced, and two metrics for quantifying coupling were referenced as ways to compare alternative design solutions for similar problems. Powerful tools in the field of Natural Language processing, specifically deep language representation models were also identified as being relevant to the area of engineering design. At the intersection of Axiomatic Design theory and these deep neural networks, an area of applicability was discovered where design statements in the form of Functional Requirements and Design Parameters could be mapped to multidimensional vector space, and then manipulated to evaluate certain designs and compare functional coupling between various solutions.

For simple cases where the semantic similarity between respective FR-DP pairs was unambiguous, the computed metrics for functional coupling aligned with expectations for these design cases. While the results of these experiments were sometimes in agreement with AD theory, in the case where coupling was not explicitly apparent to the language model used, the uncoupled design was evaluated as being insignificantly different from the coupled case. For design cases where functional independence is a result of isolating a design parameter based on a law of nature with which the sentence representation model is not familiar, the coupling metrics prove not to be as meaningful. This limitation highlights the need for further fine-tuning of the models to ensure that laws of science are passed to the model during training. This work provides a starting point for further research leveraging the power of deep learning tools in the field of systems design.

Acknowledgements

The authors would like to thank MIT/Sensetime mission-driven project award and NSF LEAP HI Award 1854833 for financial support.

References

1. Suh, N. P. (1990). The principles of design (No. 6). Oxford University Press.
2. 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.
3. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.
4. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
5. 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).
6. 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.
7. 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).
8. 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.
9. Rinderle, J. R. (1982). Measures of functional coupling in design (Doctoral dissertation, Massachusetts Institute of Technology).
10. Suh, N. P., & Rinderle, J. R. (1982). Qualitative and quantitative use of design and manufacturing axioms. Cirp Annals, 31(1), 333-338.
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
12. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771.
13. Suh, N. P. (1995). Axiomatic design of mechanical systems. American Society of Mechanical Engineers Digital Collection.