Proceedings of the ASME 2021
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC-CIE2021
August 17-19, 2021, Virtual, Online

DETC2021-66898

AUTOMATING DESIGN REQUIREMENT EXTRACTION FROM TEXT WITH DEEP LEARNING

Haluk Akay, Maria Yang, Sang-Gook Kim
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA

ABSTRACT

Nearly every artifact of the modern engineering design process is digitally recorded and stored, resulting in an overwhelming amount of raw data detailing past designs. Analyzing this design knowledge and extracting functional information from sets of digital documents is a difficult and time-consuming task for human designers. For the case of textual documentation, poorly written superfluous descriptions filled with jargon are especially challenging for junior designers with less domain expertise to read. If the task of reading documents to extract functional requirements could be automated, designers could actually benefit from the distillation of massive digital repositories of design documentation into valuable information that can inform engineering design. This paper presents a system for automating the extraction of structured functional requirements from textual design documents by applying state of the art Natural Language Processing (NLP) models. A recursive method utilizing Machine Learning-based question-answering is developed to process design texts by initially identifying the highest-level functional requirement, and subsequently extracting additional requirements contained in the text passage. The efficacy of this system is evaluated by comparing the Machine Learning-based results with a study of 75 human designers performing the same design document analysis task on technical texts from the field of Microelectromechanical Systems (MEMS). The prospect of deploying such a system on the sum of all digital engineering documents suggests a future where design failures are less likely to be repeated and past successes may be consistently used to forward innovation.

Keywords: Design Automation, Design Representation, Functional Reasoning, Neural Networks, Product Development, Product Design

1. INTRODUCTION

From customer interview transcripts, to 2D sketches and 3D models, to component manufacturing and final assembly, to legal documentation protecting IP, to published reports and papers describing performance; massive amounts of information are generated during the product development process. Over the past two decades, design and manufacturing processes have been digitalized so thoroughly that now more than 10^9 Terabytes of new industrial data is generated every year [1]. While this explosion in available Big Data has proven especially instrumental for certain Machine Learning based fields, others, including engineering design, have experienced the *curse of dimensionality*: the overwhelming number of dimensions, or data attributes and features, needed to be considered in order to extract useful knowledge [2]. Too often, the data resulting from smart manufacturing and digitalization of design practice disappears into archives because the task of combing through design documentation to identify, extract, and structure functional requirements is too monumental for human designers to perform themselves on vast amounts of data. Functional requirements (FRs) are *what* a design must achieve, and the set of FRs for a design opportunity define the functional domain of the design. *How* these FRs are satisfied is up to the designer, who may choose design parameters (DPs), which are essentially physical design solutions which address the FRs of a problem. Each *what-how* pair may spawn new FRs needing to be addressed, resulting in a large design hierarchy for complex problems. Automatically identifying the highest-level FR and extracting the underlying functional structure is the key motivation of this work.

The notion of Functional Requirements cuts across multiple approaches to thinking about design. In Systematic Engineering Design [3], establishing a functional structure with the goal of identifying the overall function of a system is the paramount goal of design practice. In Axiomatic Design thinking [4] highest-level FRs must be identified prior to mapping function to the physical domain where design parameters are established to define designed solutions satisfying requirements. In Product Design and Development [5] the identification and decomposition of customer/user needs is the critical initial step in product planning. For this work, we will refer to the definition of a Functional Requirement from Axiomatic Design as “the minimum set of independent requirements that completely characterize the design objectives for a specific need” [4].

If documentation from past designs could be automatically processed to accurately extract functional requirements, the resultant database would be invaluable for guiding product design practice. Analogous to how coding libraries of pre-tested subroutines enable good software design, extracted functional requirements from past designs can be used to guide early-stage product design. Often detailed legacy design documentation is available but difficult to digest by junior designers or those without familiarity in a specialized sub-domain. An example is Microelectromechanical Systems (MEMS) design and fabrication, which involve solving complex problems with many functional requirements at high prototyping cost, low batch yields, and long lead times. If such design processes could be informed by automatically processed past documentation, innovation may be accelerated in these fields.

While this paper describes a fully automated process requiring no human intervention, this work is a part of our effort to pursue the paradigm of *Hybrid Intelligence*, where repetitive design tasks requiring deep memory and computational power are automated by Machine Learning (ML) based methods to process data which may interactively inform the creative work of human designers [6]. An important factor to consider when applying ML to design processes is usability. Despite recent advances in artificially intelligent systems, it is impossible to perfectly replicate the analytic work that an experienced human designer can perform to distill documentation. However, the time and cost of dedicating experienced professionals to repetitive tasks can be significant. If an ML-based system can automate a repetitive task, such as processing design documentation to extract functional requirements, above an acceptable accuracy threshold, at a fraction of the time and cost, then the benefit of instantly accessing orders of magnitude more structured information justifies this application of technology to design.

This research applies ML models, specifically in the domain of Natural Language Processing (NLP), to automate the task of extracting a hierarchical structure of key functional requirements from long-form textual design documentation. Based on the assumption that valuable design information exists in a document, the method proposed in this paper will extract it. Abstracts of research papers describing designs should contain at least one high-level FR as well as *how* this FR is addressed. This is not representative of design document artifacts from industry but provides a rich test-bed for our extraction method. To validate this work, excerpts taken from published papers describing MEMS designs are automatically decomposed and key functional requirements are identified using this method. The NLP-automated results are evaluated against the judgement of a human subject-matter expert and compared to the baseline performance of 75 participating engineering designers completing the same task on the same paper excerpts.

2. BACKGROUND

In this work, design documentation in textual form is primarily considered for automating the task of extracting functional requirements. The core operations of the system presented are performed using the advanced language representation model “BERT” developed by Google AI in 2018 [7]. Therefore, this section provides background in the field of Natural Language Processing (NLP), and applications of NLP in engineering design are overviewed.

2.1 ML-Based Natural Language Processing

A key goal in NLP is to represent language quantitatively. In 2003, a seminal paper [8] was published describing a probabilistic framework for effectively converting words to multi-dimensional vectors encoding semantic meaning, which was quickly implemented using neural networks by a number of academic and NLP industry research organizations. As neural network architectures grew more sophisticated, so did the capabilities of the language models they trained. In 2017, researchers at Google unveiled a novel neural network architecture which was dubbed the *Transformer* [9] and demonstrated how sequences could be processed using Attention. The Attention mechanism is a method by which an ML model can consider a language sequence intelligently by giving more weight or “paying attention” to more contextually relevant words.

Based on the Transformer network architecture, in 2018 Google AI released the language model *Bidirectional Encoder Representations from Transformers* or BERT [7]. In addition to outperforming other existing language models in benchmark NLP metrics at the time of its release, BERT was specifically designed to be accessible by the scientific community. The bulk of computationally intensive pre-training, which allows BERT to learn word meaning and sentence context over a massive dataset of 3.3 billion words, is de-coupled from a second “fine-tuning” training phase, typically requiring a dataset of about only 10^5 examples, which fine-tunes model parameters on a specific task. This allows users of BERT to benefit from the performance of a highly trained model (pre-trained by Google), while retaining the flexibility of adapting its function (by fine-tuning with a manageable dataset) to address a specialized NLP task. Such applied tasks vary based on use, and may include forms of sentiment analysis, text summarization, and translation. The task primarily applied in this work is that of question-answering. The benefit of applying an AI-based model for this task is that in practice, FRs and DPs stated in design documentation may not necessarily strictly follow theoretical formats, i.e., “verbs” for requirements and “nouns” for design parameters. NLP models trained on language comprehension do not rely on part-of-speech tagging or keyword searching to perform extraction, which is detailed in the following subsection.

2.2 Question-Answering with BERT

Question-answering is a fundamental information retrieval task in NLP, with well-defined inputs and outputs. Simply, given context and a question, the answer must be identified within the given context, if an answer exists. In the case of *extractive* question-answering, the answer is extricated, unmodified, as a single sub-sequence (or *span*) from the context. Therefore, if we consider a context containing N number of words, the answer can be defined as the span of words from the i^{th} word to the j^{th} word where $i \leq j \leq N$.

$$f(Q,C) = [i,j] \tag{1}$$

The task of extractive question-answering can then be modeled as a function f with two textual inputs and two numerical outputs, as shown in expression (1). The first text input is the "question" span Q, and the second is the "context" span C. The two numerical outputs are the indexes demarcating the answer span within C with the indices i (start index) and j (end index). The initial pre-training step of BERT results in a Transformer Neural Network (*TransNN*) with trained parameters which can take in a word and output a D dimensional vector encoding the word's semantic information. If we consider the question as a sequence containing M words $Q = (q_1, q_2, \ldots, q_M)$ and the context similarly containing N words $C = (c_1, c_2, \ldots, c_N)$, then these word sequences may be encoded into vector arrays $\mathbf{Q} \in R^{M \times D}$ and $\mathbf{C} \in R^{N \times D}$ in a feature space of D dimensions, as illustrated in expression (2). For the BERT model utilized in this work, D = 1024.

$$\begin{aligned} \mathbf{Q} &= \mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_M} = TransNN(q_1, q_2, \ldots, q_M) \\ \mathbf{C} &= \mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_N} = TransNN(c_1, c_2, \ldots, c_N) \end{aligned} \tag{2}$$

As previously stated, following the initial pre-training step where the model essentially learns to convert words into vectors, there is a fine-tuning step requiring a new task-specific dataset. For question-answering, one of the largest and most well-curated datasets is the Stanford Question Answering Dataset (SQuAD) [10], which is a set of 100,000 crowd-sourced examples of contexts, questions, and correct answers. SQuAD examples are generally nontechnical and cover a wide range of topics, meaning the model may be applied to various design topics, but there is opportunity to further fine-tune on domain specific literature if a single design domain is of particular interest.

For the specific case of fine-tuning BERT on SQuAD to perform the task of question-answering, the developers of BERT introduce two new vectors $S \in R^D$ and $E \in R^D$, the elements of which are learnable parameters. While iterating through the question-context-answer examples, the elements in the new vector S are optimized such that when the dot product between S and any vectorized word from the context sequence $\mathbf{c_i}$ is taken, a measure of likelihood of that particular word being the start of the answer, is returned. The same is true for the new vector E being trained to identify the end position of the answer. The exact probabilities are found by normalizing exponentially over all other N word vectors in C, as shown in the equations in (3).

$$\begin{aligned} P_i &= \frac{e^{S \cdot \mathbf{c_i}}}{\sum_{k=1}^{N} e^{S \cdot \mathbf{c_k}}} \\ P_j &= \frac{e^{E \cdot \mathbf{c_j}}}{\sum_{k=1}^{N} e^{E \cdot \mathbf{c_k}}} \end{aligned} \tag{3}$$

The answer span is then identified by the pair of indices $[i, j]$ with the highest summed probability where $j \geq i$. BERT, fine-tuned on SQuAD, is capable of performing question-answering rivaling human reading comprehension, and if the correct inputs are used, may be applied to engineering design to help extract functional information given context.

2.3 NLP for Automating Requirement Extraction

While some creative steps in design are a true artistic craft which only humans are capable of executing, other steps are repetitive, painstaking and prone to human error. Hybrid Intelligence in design [6] is a model for collaboration between human designers and machines where critical tasks, which may benefit from the computational power of machines able to process vast amounts of data, are automated with ML-based models. The contributions of this work adhere to this principle and follow a path of research applying NLP to distill design requirements automatically from documentation. In previous work [11], we characterized designed systems in terms of functional coupling solely based on textual descriptions of their functional requirements (FRs) and associated physical solutions designed to address a given functional requirement. This research demonstrated that the semantic domain of language is mirrored by the functional domain of design, i.e. that word meaning similarity could be used to approximate the degree to which certain FRs may be affected by a given DP. By obtaining vector representations of succinct design descriptions from pre-trained neural networks, similar to those in the expression in (2), measures of functional independence in simple systems could be accurately quantified.

Given the demonstrated feasibility of applying ML methods to process design documentation, we have developed a process [12] for helping designers read design documentation by applying clustering methods to structure design-related information extracted using the information retrieval function of BERT, fine-tuned on SQuAD. By embedding short, extracted spans with NLP-based representation methods into feature vectors, a large pool of such spans from a variety of different documents describing the same design opportunity could be structured by clustering the vectors based on semantic similarity. In this way, a hierarchy of functional requirements could be surfaced from long-form documentation.

3. METHOD

The problem being addressed can be described as follows. Given a text passage of around 300-500 words, identify all the functional requirements (FRs) and design parameters (DPs) explicitly stated in the passage, and extract a structured hierarchy of these FRs and DPs as a representation of the design, from the text narrative. The system for extracting and structuring FRs and DPs from text introduced in this work is based on a recursive algorithm for decomposing designs inspired by the interplay between the functional and physical domains in principles of Axiomatic Design [4]. This section is divided into two subsections detailing the recursive algorithm and demonstrating its use with a case study with documentation from Microelectromechanical System (MEMS) design. MEMS design processes are well-documented in publicly available literature, address complex design problems with many FRs, and require specialized domain knowledge for gaining expertise. Although in industry, design documentation is often less structured and more succinct, the reason for using published paper abstracts was the expectation that well-structured paper abstracts should densely contain functional requirements that would provide a rich context for demonstrating information extraction.

3.1 Method of Extracting Functional Requirements

The method for extracting functional requirements (FRs) from a text passage describing a design is based on the assumption that a hierarchy of FRs exists, with the "root node" or highest-level FR defining the overarching aim of the design. The objective of this method is to identify all information in the passage which may be relevant to defining FRs in such a hierarchy, decomposing thoroughly from the top-down until all the lowest-level "leaf nodes" have been identified. The method implemented to extract such a hierarchical structure is a form of tree traversal, initialized by identifying the "root node" highest-level FR and decomposing downwards. Extractive question-answering is implemented recursively, with the input question Q and context C continuously updating with every extracted FR. The following subsections describe this method in detail.

3.1.1 Identifying the Highest-Level FR

The tool used for extracting functional requirements (FRs) is Google AI's language model BERT fine-tuned for extractive question-answering on SQuAD. In order to obtain the indices [i, j] accurately demarcating the position of the FR of interest within the context C, the correct question Q must be posed. The case of identifying the highest-level FR, which can be denoted as FR_0, poses a unique challenge because, in accordance with the top-down strategy of decomposition, no information has yet been extracted which might have been used as referential material. The question chosen to elicit FR_0 from context to initialize the top-down decomposition is a simple "What"-type query prompting the return of the most overarching design goal, such as Q_0 shown in (4).

$$Q_0 : \texttt{What is the aim?} \tag{4}$$

The wording of this question was determined after experimenting with various synonyms and phrasing choices in a previous study [12] based on the definition of functional requirements describing "What, not How," [5] and embodying "What we want to achieve" [4]. It was found that, in order to elicit the highest-level FR which is expected to encompass the entire functional domain of the design, such a broad generalized question was needed. Q_0 can be generalized to initialize the top-down decomposition for any design document by supplying multiple question permutations using synonyms, and identifying FR_0 based on the maximum confidence score returned by the model.

3.1.2 Recursion for Extracting Hierarchical Structure

The method by which the remaining functional requirements (FRs) are identified is a form of tree traversal, which started with the highest-level FR_0. The hierarchical structure is extracted in a top-down approach. At each structural level, there exists a discrete number of nodes, each of which contains one FR, or a "what" of the design. Any number of these FRs may be paired with a design parameter (DP), or "how" the design addresses the "what" defined by the given FR. Hierarchically superior to all the nodes on this level must exist a node containing a higher-level FR-DP pair, as it is this completed "what-how" combination which is able to be decomposed into the FRs existing in the nodes on each subsequent level. In a fully defined design, every "what" (FR) is addressed by a "how" (DP), but when extracting such a structure from documentation, incomplete functional information is natural and expected. Where FRs are not defined or explicitly addressed by a DP in documentation, the decomposition along that node is terminated, as shown in Figure 1.

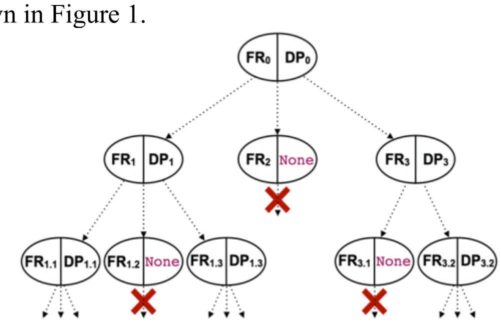

FIGURE 1: Example hierarchical structure of the functional domain, to be extracted from textual documentation

Apart from FR_0, which has no superior and is identified as detailed in section 3.1.1, every other FR can be identified using question-answering where the same question format is recursively posed. This question contains the "what-how" information of the previous superior node, which are termed FR_{sup} and DP_{sup} in the expression below.

$$Q_{FR} : \texttt{What is needed for } \{DP_{sup}\} \texttt{ to } \{FR_{sup}\}\texttt{?} \tag{5}$$

Following the identification of each FR on a given structural level, question-answering can be used to identify an associated DP addressing that FR using the query expressed below.

$$Q_{DP} : \texttt{How does it } \{FR\}\texttt{?} \tag{6}$$

If a DP addressing the FR does not exist in the context, a null answer is returned. In such instances, that lone FR node cannot be further decomposed. After each FR-DP pair is identified (or at least queried), the context C is updated by removing any retrieved content so that the next time the question (5) is prompted, a new result will be returned. The decomposition of the superior node is terminated when the answer to (5) posed to the updated context is null, and the process continues for each FR-DP pair identified. This recursive method is ultimately terminated when either all yielded answers are null, or the context updates have resulted in an empty sequence C from which no more information may be extracted. The full process is detailed with pseudo-code in Table 1, and results in an extracted tree of structured functional requirements pertaining to a single key highest-level FR of the design.

Table 1: Design Decomposition Algorithm

```
Input:
  Context C_0 (text document describing design)
Definitions:
  Q_0 : What is the aim?
  Q_FR : What is needed for {DP_sup} to {FR_sup}?
  Q_DP : How does it {FR}?
BERT-QA:
Function which takes a context C and a question Q as inputs, and extractively
returns an answer span as a subsequence of C
INITIALIZE:
  FR_0 = BERT-QA (C_0 , Q_0)
  if FR_0 is None then:
    terminate
  else
    DP_0 = BERT-QA (C_0 , Q_DP{FR_0})
    if DP_0 is None then:
      output FR_0
      terminate
    else
      C_1 = C_0 – (FR_0 and DP_0)
      output FR_0 , DP_0 , C_1
      Goto DECOMPOSE
    end
  end
DECOMPOSE:
  Inputs: FR_0 , DP_0 , C_1
  i = 1
  while FR_i is NOT None :
    FR_i = BERT-QA (C_i , Q_FR{ FR_0 , DP_0})
    DP_i = BERT-QA (C_i , Q_DP{FR_i})
    C_i+1 = C_i – (FR_i and DP_i)
    Output FR_i , DP_i , C_i+1
    i = i + 1
  when FR_i is None :
    for every (FR_i , DP_i) pair that is NOT None :
      if C_i+1 is empty:
        terminate
      else:
        recursively repeat DECOMPOSE (FR_i , DP_i , C_i+1)
      end
    end
  end
```

3.2 MEMS Case Study

In order to demonstrate the operation of the automated system for extracting a hierarchical structure of functional requirements (FRs) from textual design documentation, a case in the field of Microelectromechanical Systems (MEMS) design is chosen. The primary reason for selecting this field is the scientific nature in which design specifications are documented through publications which can be expected to be rich with the "what" (FRs) and "how" (DPs) being sought by our system. Published papers in this field provide us with real textual design data with which to run such a case study and publicly share results. However, the eventual aspiration of this research is to deploy such an automated system on larger sets of (possibly proprietary) design data so that designers in industry may easily benefit from decades of documented design efforts.

3.2.1 Low-Frequency Vibrational Energy Harvesting

This design problem in MEMS Energy Harvesting seeks to bridge a gap between a real-world opportunity, and the limitations of physics. In remote or mobile environments, sources of electric power can be scarce, but ambient vibrations naturally occurring in the surroundings may be harvested and converted to usable electric power. For such cases, energy harvesting systems employing the properties of piezoelectric materials may be applied. The piezoelectric effect is exhibited by special materials which, when experiencing mechanical deformation, accumulate electric charge which may be stored and converted into usable electric power. This is a highly applicable feature for vibrational energy harvesting but requires repeated straining of piezoelectric materials in order to work.

The key design challenge for vibrational energy harvesting via micro-scale piezoelectric structures is that ambiently occurring vibrations are generally low frequency (below 100Hz), while natural linear resonance scales inversely with size [13-14]. As illustrated in (7), natural resonance ω_0 can be expressed as a function of stiffness k and mass m, where the micro-scale dimensions L of MEMS piezoelectric beams can be related to very small masses through density P as shown in (8).

$$\omega_0 = \sqrt{\frac{k}{m}} \qquad\qquad m = PL^3 \qquad (7)\,(8)$$

For background, common types of FRs for MEMS vibrational energy harvesters define system goals which include the requirements which the design must meet to operate and perform the desired functionality, such as "harvesting energy" or "resonating at [desired] frequency range." For piezoelectric energy harvesting specifically, DPs satisfying the identified FRs may detail how material deformation may produce energy, and how geometric structures may enable resonance at a target frequency range. This specific low-frequency vibrational energy harvesting design problem can be addressed by eschewing linear resonance as the primary means of exciting a dynamic response in the piezoelectric material, and instead relying on nonlinear methods to generate strain from low frequencies at the microscale.

3.2.2 Results from Automated FR Extraction

Following the algorithm detailed in Table 1, two abstracts from published papers were taken as the input context C_0 and

FRs were extracted starting with the highest-level FR_0. The results of this experiment are presented by showing the identified FRs in context in Figure 2, as well as the extracted "What" (FR) – "How" (DP) structure in Table 2.

Table 2: Extracted "What" (FR) – "How" (DP) Structure

Extracted Structure from Abstract 1 [13]

	"What"		"How"
FR_0	**lowering the operating frequency window below 100hz for the first time at the mems scale**	DP_0	**bistable buckled beam energy harvester**
FR_1	buckling of 200 μm	DP_1	bistable
$FR_{1.1}$	progressive residual stress control of the deposition processes along the fabrication steps	$DP_{1.1}$	multilayer beam structure with residual stress induced buckling
$FR_{1.2}$	input energy overcomes an energy threshold	$DP_{1.2}$	large snapping motion of the beam at very low frequencies
$FR_{1.3}$	50 % bandwidth under 70hz at 0 .5g input	$DP_{1.3}$	operating conditions that have not been demonstrated by mems vibration energy harvesters before
$FR_{1.1.1}$	vibration energy harvesters based on the resonance	$DP_{1.1.1}$	work effectively only when the operating frequency window of the beam resonance matches with the available vibration source
$FR_{1.3.1}$	none of the resonating mems structures can operate with low frequency, low amplitude, and unpredictable ambient vibrations	$DP_{1.3.1}$	the resonant frequency goes up very high as the structure gets smaller

Extracted Structure from Abstract 2 [14]

	"What"		"How"
FR_0	**harvests energy from parasitic ambient vibration**	DP_0	**piezoelectric effect**
FR_1	bending strain	DP_1	None
FR_2	robust power generation	DP_2	wide bandwidth of resonance enables a robust power generation amid the uncertainty of the input vibration spectrum
FR_3	passive feedback and consequently a wide-band resonance	DP_3	stiffness nonlinearity due to the stretching
FR_4	tensile stretching strain	DP_4	in doubly-anchored beams
FR_5	low power density	DP_5	prevents them from practical use
FR_6	ultra wide-bandwidth	DP_6	None
FR_7	nonlinear	DP_7	wide-bandwidth
FR_8	power density	DP_8	comparing the frequency response of the system with that of an equivalent linear harvester with a similar q - factor

Abstract 1, from [13]

Vibration energy harvesters based on the resonance of the beam structure work effectively only when the operating frequency window of the beam resonance matches with the available vibration source. None of the resonating MEMS structures can operate with low frequency, low amplitude, and unpredictable ambient vibrations since the resonant frequency goes up very high as the structure gets smaller. Bistable buckled beam energy harvester is therefore developed for **lowering the operating frequency window below 100Hz for the first time at the MEMS scale**. This design does not rely on the resonance of the MEMS structure but operates with the large snapping motion of the beam at very low frequencies when input energy overcomes an energy threshold. A fully functional piezoelectric MEMS energy harvester is designed, monolithically fabricated, and tested. An electromechanical lumped parameter model is developed to analyze the nonlinear dynamics and to guide the design of the nonlinear oscillator based energy harvester. Multilayer beam structure with residual stress induced buckling is achieved through the progressive residual stress control of the deposition processes along the fabrication steps. Surface profile of the released device shows bistable buckling of 200µm which matches well with the amount of buckling designed. Dynamic testing demonstrates the energy harvester operates with 50% bandwidth under 70Hz at 0.5g input, operating conditions that have not been demonstrated by MEMS vibration energy harvesters before.

Abstract 2, from [14]

An ultra wide-bandwidth resonating thin film PZT MEMS energy harvester has been designed, modeled, fabricated and tested. It **harvests energy from parasitic ambient vibration** at a wide range of amplitude and frequency via piezoelectric effect. At the present time, the designs of most piezoelectric energy devices have been based on high-Q linear cantilever beams that use the bending strain to generate electrical charge via piezoelectric effect. They suffer from very small bandwidth and low power density which prevents them from practical use. Contrarily, the new design utilizes the tensile stretching strain in doubly-anchored beams. The resultant stiffness nonlinearity due to the stretching provides a passive feedback and consequently a wide-band resonance. This wide bandwidth of resonance enables a robust power generation amid the uncertainty of the input vibration spectrum. The device is micro-fabricated by a combination of surface and bulk micro-machining processes. Released devices are packaged, poled and electro-mechanically tested to verify the wide-bandwidth nonlinear behavior of the system. Two orders of magnitude improvement in bandwidth and power density is demonstrated by comparing the frequency response of the system with that of an equivalent linear harvester with a similar Q-factor.

Key: Functional Requirement **Highest-level FR_0**

FIGURE 2: Functional requirements automatically identified, highlighted in context from [13-14]

4. HUMAN DESIGN EVALUATION: SURVEY STUDY

In order to evaluate the performance of this automated NLP-based system, we may compare the results to the judgements of a MEMS expert with first-hand knowledge of the designs documented in each abstract. The principal investigator of the group where the research [13,14] was performed provided the "ground truth" for the highest-level requirements and solutions (FR_0 and DP_0) in each design. This human Subject-Matter Expert (SME) judgement is used to evaluate the accuracy of the NLP performance on identifying the highest-level design information.

While the "ground truth" established from the SME is necessary to measure accuracy, we are also interested in comparing the NLP-based results to a "baseline" measure of how manual analysis of design documentation is currently performed.

The baseline was established by surveying 75 human designers, with varying degrees of familiarity in MEMS, who performed the same design decomposition tasks on the two passages. In this section, the ground truth from the SME and the baseline results from the survey are used to evaluate the NLP-based system's performance.

4.1 Survey Design

The survey was implemented using Qualtrics XM software. The respondents were designers of varying levels of experience from the MIT community. The survey was distributed to graduate researchers, research scientists, and faculty members in the MEMS research community at MIT, and to undergraduate, graduate, post-doctoral, and faculty members of the Mechanical Engineering department, and also to graduate students in the Systems Design Management program. Participation in the survey was incentivized by the chance to win prizes in a raffle.

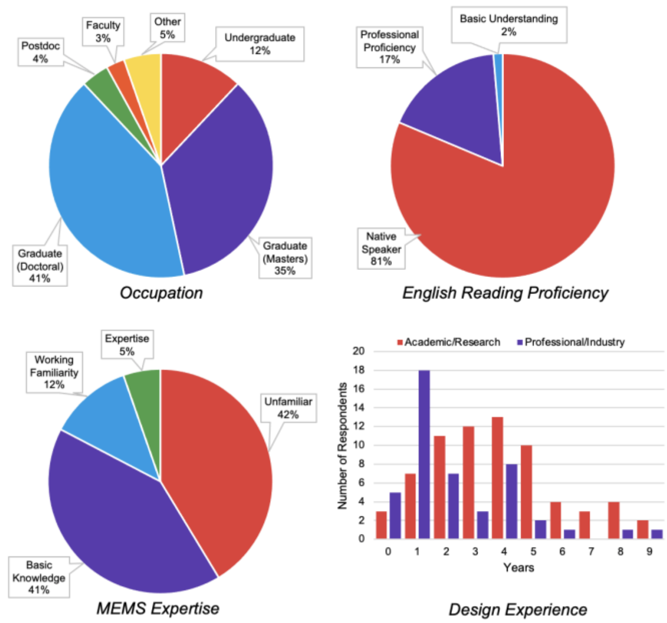

FIGURE 3: Demographic information of survey participants

The first part of the survey collected demographic identifiers shown in Figure 3. The second section of the survey trained the respondents on the software's user interface for labeling text spans, and also established an understanding of the definition of functional requirements and a top-down hierarchical method of extracting FR structure starting with the highest-level FR_0. This consisted of three examples with easily recognizable FRs and DPs (which were termed "requirements" and "solutions" respectively for survey respondents), as well as obvious examples of extraneous information. Respondents were able to highlight words and choose from suggested labels what type of information they were identifying. Following each trivial task, the "correct" answers were shown to the respondent to reinforce the goals of the training exercise. In the third section of the survey, the two MEMS design texts from the case study in Section 3 were shown to respondents for analysis. Respondents were asked to label just one highest-level FR_0 and DP_0, and then subsequently to also label as many other FRs and DPs as they could identify, using the same tools and definitions learned during the training exercises.

4.2 Survey Results

In total, 75 respondents completed the entirety of the survey questions out of a total of 165 individuals who started the survey. The average time taken to complete the design decomposition tasks were 5:56 and 5:35 minutes for the abstracts 1 and 2. Respondents completed design decompositions of the MEMS abstracts by using the highlighting tool from the training exercises to first identify one highest-level requirement (FR_0) and then one highest-level solution (DP_0). Based on this highest-level information, they next identified any other FRs which they believed to be requirements for the design defined by the high-level FR_0-DP_0 they initially selected. We can compare the NLP model-based results to the judgements of the SME and human designer results by visualizing the number of human "votes" for where these selections were made in the passage context. In Figures 4-5, the distribution of these votes is plotted with respect to the indexed sequence of words from the MEMS context for the highest-level FR_0 and DP_0 selections for both passages, alongside their textual form for a complete visualization. The most popular spans, where most respondents indicated a high-level "what" or "how" occurred and the span selected by the NLP model and the SME are indicated.

The second part of the respondents' task was to identify the lower-level functional requirements (FRs) stated in each passage, shown in Figures 6-7. Because in this case, the task involved identifying multiple FRs, a representative fragment selection of 12 of the 75 total individual respondent's labels is also displayed (bottom-left of each figure) to illustrate a typical individual designer's analysis of the text. For the selected individual responses, 3 individuals from each of the four categories of MEMS expertise are displayed. On average, each individual designer selected between 3 and 4 FRs for each passage, while the NLP-based system identified 7 and 9 respectively. For such an information retrieval task, commonly used metrics for performance are precision and recall. Precision measures how many selected FRs are actual FRs (ratio of true positives to true positives plus false positives). Recall measures how many actual FRs are selected (ratio of true positives to true positives plus false negatives). The baseline comparison used for calculating these metrics for the precision and recall of lower-level FRs was established according to the consensus of all the survey respondents. The local maxima exceeding a threshold of 5 votes were used to determine the peaks which indicated the baseline FRs. The resulting scores are shown in Table 3.

Table 3: Precision and Recall Scores of NLP-based system

	Precision	Recall
Abstract 1	0.71	0.55
Abstract 2	0.89	0.89

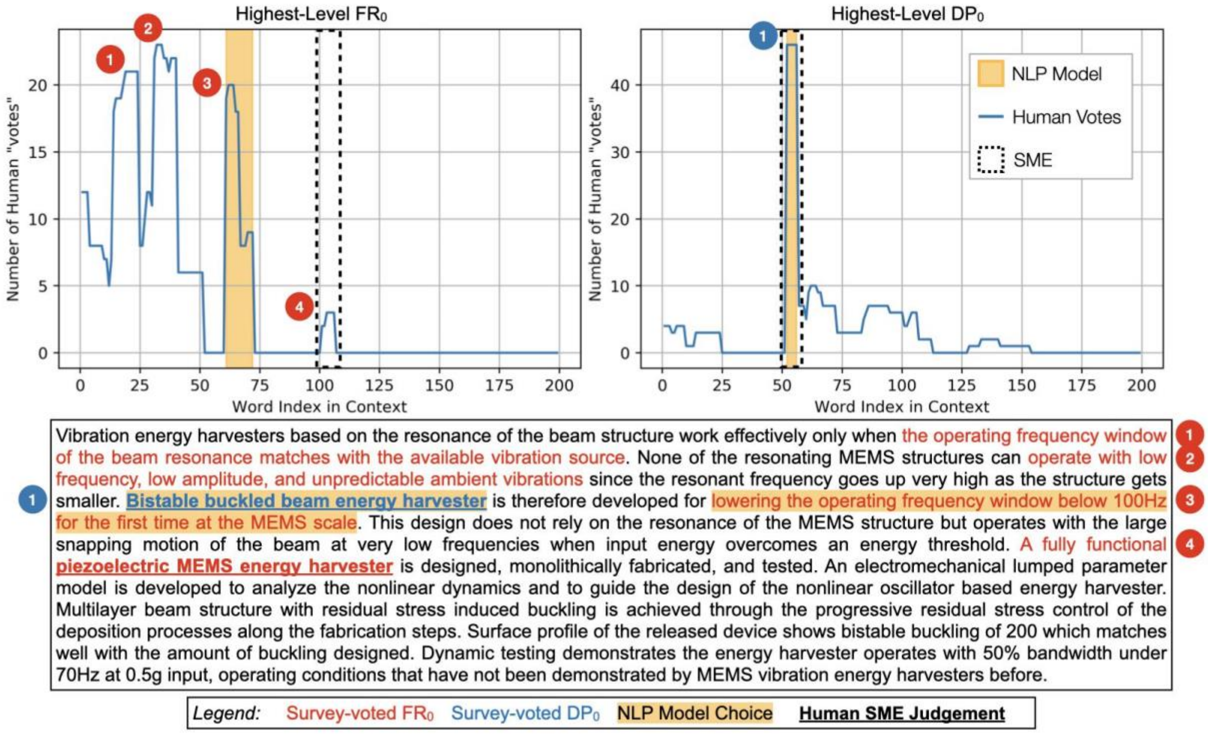

FIGURE 4: Comparison of highest level "what" (FR_0) and "how" (DP_0) choices of NLP-based system, Human Designers Surveyed, and the Subject-Matter Expert (SME) for Abstract 1 [13]

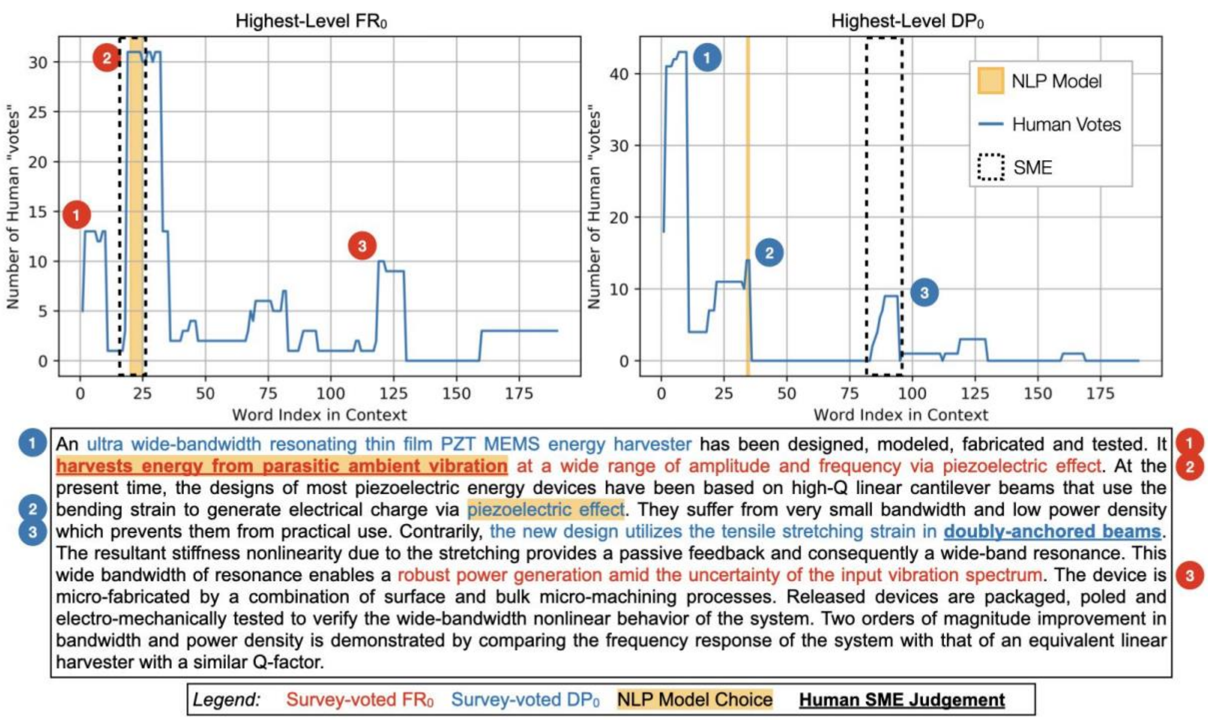

FIGURE 5: Comparison of highest level "what" (FR_0) and "how" (DP_0) choices of NLP-based system, Human Designers Surveyed, and the Subject-Matter Expert (SME) for Abstract 2 [14]

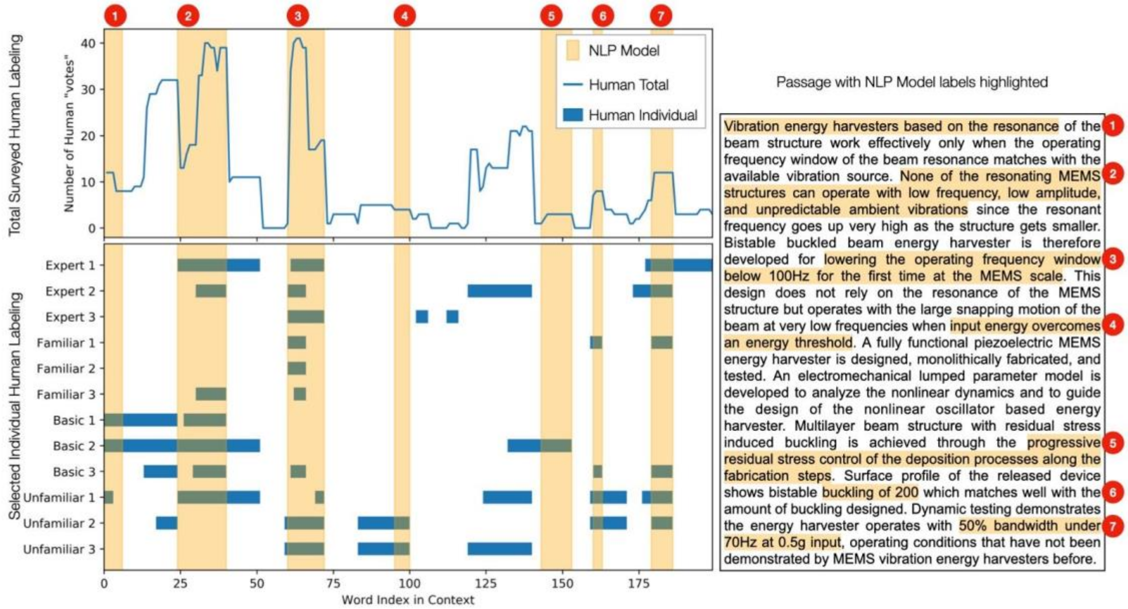

FIGURE 6: Comparison of Functional Requirement labeling between selected human individuals (bottom), human aggregate (top) and NLP-based system (highlights) for Abstract 1 [13]

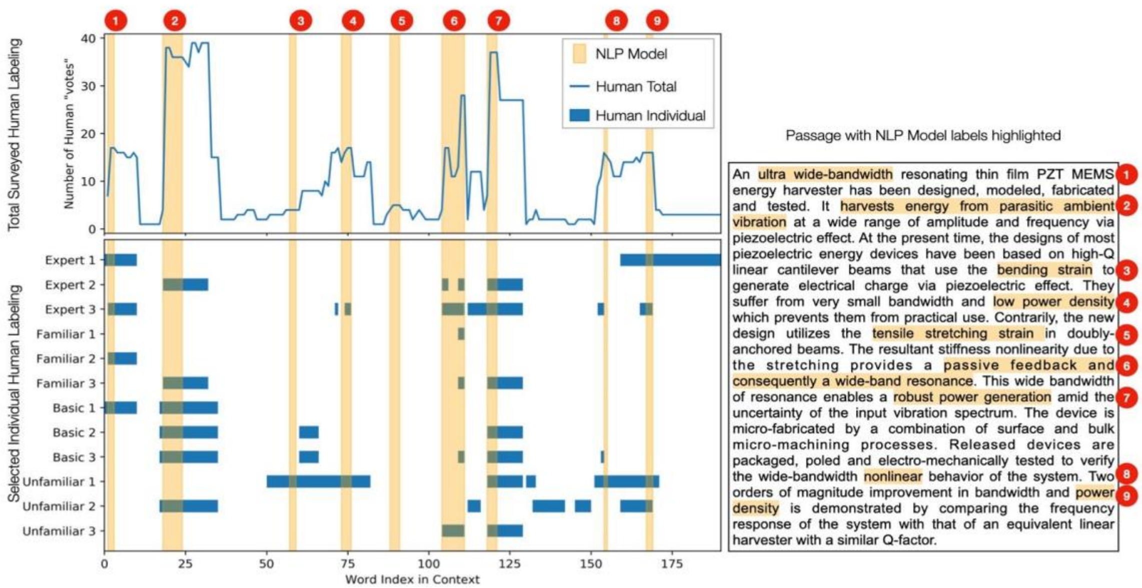

FIGURE 7: Comparison of Functional Requirement labeling between selected human individuals (bottom), human aggregate (top) and NLP-based system (highlights) for Abstract 2 [14]

5. DISCUSSION

The performance of the NLP-based system on the primary task of processing a text passage and returning the highest-level "what" and "how" are generally in agreement with consensus of the 75 human designers surveyed. In Figure 4, we observe that the consensus for the highest-level FR is divided among 3 different peaks which relate to semantically similar spans of text all describing the requirement of lowering the operating frequency window for the designed device. The NLP selection is one of these 3 peaks, while the SME judgement settled for a fourth span in the text. For the highest-level DP of the first passage, there is clear agreement between the human consensus, the SME, and the NLP selection. In the second passage shown in Figure 5, there is agreement between the survey consensus identifying the highest-level FR and the NLP selection. For the highest-level DP, neither the SME nor the NLP selection point to the human popular consensus. Because the method by which the "how" (DP) is mapped from the identified "what" (FR) involves identifying literally how the FR is addressed by the design (as detailed in Table 1), the NLP model chooses a span from the same sentence as the highest-level FR as this is where this information is explicitly stated, despite not correctly describing the highest-level design parameter in the passage.

When evaluating the performance of the NLP-based system for extracting all FRs contained in each passage, the precision and recall metrics indicate higher performance on the second passage (0.89 for both precision and recall) than the first passage (0.71 precision and 0.55 recall). It is noted that the NLP-based system extracted more FRs (9) for the second passage than for the first (7) before the algorithm terminated, suggesting that the full hierarchical tree structure of design information may not have been completely traversed when analyzing the first passage. Individual humans identified between 3 and 4 FRs on average for each passage, still suggesting a more thorough traversal of the design tree by the model when compared to a given individual, but not perfect when compared to the consensus of 75 designers. It is also noted that humans averaged 5:56 and 5:35 minutes for analyzing each passage respectively, while the execution of the pre-trained NLP-based system took approximately 10 seconds on a common CPU. The comparison with the survey suggests the NLP-based system accurately identifies highest-level functional requirements. This discrepancy between the SME and survey results also reinforces the difficult nature of this decomposition task.

One family of designs where multiple DPs may address a single FR, is uncommon after the product development cycle, but the algorithm may easily be modified to do so by repeating the "How"-type question-answering search for the same FR to extract multiple DPs, and recurrently pursue these new extracted branches.

This work focuses on extracting design information from free text and was tested on passages taken from peer-reviewed academic literature which documented design. In industry, however, documentation can be less structured and more terse. While NLP models, such as those used in this work, have demonstrated the ability to process shorthand note-like documentation, the effort to compile enough textual data from a variety of formats including presentations, online collaborative tools, and other electronic communication used in the design and production industry, is not trivial. Before the NLP-based system presented in this paper can be deployed directly for analysis in industry, a concentrated effort to compile and curate design documentation is required. Additionally, a next step of this work is to consider how extracted design trees may be meaningfully presented to designers; a fully decomposed hierarchy may still require further abstraction to become informative. Constructing a searchable knowledge base of many structured FRs will require compiling multiple trees together for a comprehensive result.

6. CONCLUSION

Based on the opportunity provided by abundant digitalized data in industry, a system for automatically processing design documentation has been developed using models from Machine Learning-based Natural Language Processing. An algorithm utilizing recursive question-answering to traverse a hierarchical tree structure of interrelated functional requirements and design parameters was introduced. Its performance was evaluated in comparison to the analysis by a human subject-matter expert and designers of two passages documenting MEMS design via a survey study. Agreement between the NLP-based system and the human respondents suggest an opportunity for changing the way design data is processed at the industry level.

Next, this research aims to distill design documentation from every step of the production cycle to be able to relate functional requirements, design parameters, and process variables and accurately model interdependencies between these domains. The goal is to leverage data from industry in order to aid designers and maximize the probability of success for future design innovation.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (NSF) Leading Engineering for America's Prosperity, Health, and Infrastructure (LEAP HI) program, award number 1854833.

REFERENCES

[1] Yin, Shen, and Okyay Kaynak. "Big data for modern industry: challenges and trends." *Proceedings of the IEEE* 103.2 (2015): 143-146.

[2] Ur Rehman, Muhammad Habib, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, and Samee U. Khan. "Big data reduction methods: a survey." *Data Science and Engineering* 1, no. 4 (2016): 265-284.

[3] Pahl, Gerhard, and Wolfgang Beitz. *Engineering design: a systematic approach*. Springer Science & Business Media, (2013).

[4] Suh, Nam P. *The principles of design*. No. 6. Oxford University Press, (1990).

[5] Ulrich, Karl T., Steven D. Eppinger, and Maria C. Yang. "Product design and development." 7th ed. McGraw-Hill Education (2019).

[6] Kim, Sang-Gook, Sang Min Yoon, Maria Yang, Jungwoo Choi, Haluk Akay, and Edward Burnell. "AI for design: Virtual design assistant." *CIRP Annals* 68, no. 1 (2019): 141-144.

[7] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[8] Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. "A neural probabilistic language model." *The journal of machine learning research* 3 (2003): 1137-1155.

[9] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

[10] Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "Squad: 100,000+ questions for machine comprehension of text." *arXiv preprint arXiv:1606.05250* (2016).

[11] Akay, Haluk, and Sang-Gook Kim. "Design transcription: Deep learning based design feature representation." *CIRP Annals* 69, no. 1 (2020): 141-144.

[12] Akay, Haluk, and Sang-Gook Kim. "Reading functional requirements using machine learning-based language processing." *CIRP Annals* 70, no. 1 (2021)

[13] Xu, Ruize, Haluk Akay, and Sang-Gook Kim. "Buckled MEMS beams for energy harvesting from low frequency vibrations." *Research* 2019 (2019).

[14] Hajati, Arman. "Ultra Wide-Bandwidth Micro Energy Harvester." *Massachusetts Institute of Technology* (2010).