Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2015
August 2-5, 2015, Boston, Massachusetts, USA

Draft of DETC2015-47541

AUTOMATIC EXTRACTION OF FUNCTION KNOWLEDGE FROM TEXT

Hyunmin Cheong, Wei Li, Adrian Cheung, Andy Nogueira, Francesco Iorio

Autodesk Research
Toronto, ON, Canada

ABSTRACT

This paper presents a method to automatically extract function knowledge from natural language text. Our method uses syntactic rules to extract subject-verb-object triplets from parsed text. We then leverage the Functional Basis taxonomy, WordNet, and word2vec to classify the triplets as artifact-function-energy flow knowledge. For evaluation, we compare the function definitions associated with 30 most frequent artifacts compiled in a human-constructed knowledge base, Oregon State University's Design Repository (DR), to those extracted using our method from 4953 Wikipedia pages classified under the category "Machines". Our method found function definitions for 66% of the test artifacts. For those artifacts found, our method identified 50% of the function definitions compiled in DR. In addition, 75% of the most frequent function definitions found by our method were also defined in DR. The results demonstrate the promising potential of our method in automatic extraction of function knowledge.

INTRODUCTION

For CAD systems to offer capabilities beyond modeling and analyzing geometries, they inevitably require an extensive knowledge base [1]. The design research community has explored developing knowledge-based CAD systems since the early 1980's [2]. Although many systems have been developed, as reviewed in [3-5], they revealed several problems. For example, they are not able to perform creative synthesis, focused on very small domains, and posed difficulties in maintaining and updating the knowledge base [6]. In fact, the latter two problems are true to any knowledge-based systems. The usefulness and capabilities of a knowledge-based system inherently depend on the amount and quality of its knowledge.

Our research aims to investigate automatic knowledge extraction from text resources to resolve some of the challenges in enabling knowledge-based CAD systems. Recently, the artificial intelligence research community has gained significant advances in automatic knowledge base construction [7-14]. By combining research efforts in knowledge representation, machine learning, and natural language processing, automated construction of general or common-sense knowledge has become possible. Efforts are made to extract information at a large scale from the entire Web [7-9,13,14] or general knowledge sources such as Wikipedia[1] [10-12]. The goal of acquiring such knowledge is to support automated reasoning and semantic search, e.g., in Semantic Web[2]. We believe that the approaches and techniques developed for these efforts could also be used to extract domain-specific knowledge such as mechanical design knowledge. The knowledge acquired could then be used to support automated reasoning and semantic search required in knowledge-based CAD systems.

As an initiative to our research goal, we attempt to extract function knowledge from text. Several types of mechanical design knowledge are required to support knowledge-based CAD systems, such as geometric models, materials, manufacturing processes, etc. [15,16]. However, the knowledge essential for the synthesis capability is function knowledge. Designers typically an initiate design process by defining desired functions and look for solutions that satisfy them [17-19]. The concept of function is also central to many knowledge representation frameworks developed to automate and support engineering design [20-22].

In addition, significant efforts have been made to acquire function models of various electromechanical artifacts in a design repository, e.g., [23], using a controlled vocabulary called Functional Basis [24]. Such knowledge base not only aids designers in sharing and reusing their designs, but also serves as the basis for function-based synthesis tools [25-27]. Again, the usefulness of the repository and synthesis tools is directly correlated with the amount of the knowledge constructed. For instance, Kurtoglu and Campbell [26]

[1] http://en.wikipedia.org/

[2] http://www.w3.org/standards/semanticweb/

identified configuration design grammar from the existing function knowledge captured in a design repository. Acquisition of new function knowledge should help expand the grammar and increase the variety of design concepts that the method can generate.

Our method extracts function knowledge by first acquiring relation information, e.g. subject-verb-object (SVO) triplets, from unstructured text using syntactic relation rules. Then, we leverage existing function taxonomy, Functional Basis, and concept classification methods such as WordNet-based [28] word similarity measures and word2vec [29] to classify the acquired knowledge. Our current scope is focused on extracting artifact-function-energy flow knowledge from natural language text related to mechanical engineering.

The current paper is outlined as follows. Background highlights prior work in design knowledge modeling, acquisition, and application, as well as concept classification methods. Method presents how we extract and classify function knowledge from text. Evaluation reports precision/accuracy measures of our method in comparison to the knowledge compiled by humans in a design repository. The paper ends with general discussion, conclusions, and future work.

BACKGROUND

Modeling and application of design knowledge

Pioneer work in knowledge representation for design includes Gero et al. [20], Umeda et al. [21], Chandrakasen et al. [22], etc. All these representation frameworks include function as a critical element of design knowledge. In essence, an artifact is synthesized by a designer to serve a certain purpose, i.e., a function. Hence, function knowledge must be central to any design knowledge base that would be used to support synthesis tasks. For all these frameworks, prototype systems have been developed to demonstrate their synthesis capabilities within a limited problem domain, summarized in [30].

Function modeling is also established as an important early design process [17-19]. The modeling method involves defining a product's intended functions as transformation of inputs to outputs. Specifically, the method uses verbs to describe the transformation functions and nouns to describe the input and output flows. To support function modeling, Stone and Wood [31] developed a controlled vocabulary of functions and flows called Function Basis, which was later reconciled by Hirtz et al. [24]. Using this vocabulary, the Design Repository project [23] was initiated to compile the function information of various products and artifacts (i.e., components of products). So far, the design repository maintained by Oregon State University[3] (hereinafter referred to as the "DR") contains 184 products and 6906 artifacts represented in the database schema developed by Bohm et al. [32].

Several synthesis methods have been developed to leverage the knowledge compiled in the DR and support conceptual design. Bryant et al. [25] developed a concept generation method that identifies possible component chains for a given function chain. The method relies on component-function relationships and component-component compatibility relationships retrieved from the existing knowledge in the DR. Kurtoglu and Campbell [26] used a set of graph rules that identifies patterns in a function model graph and replaces sections of the graph with corresponding components. The graph grammar was developed based on observing the existing knowledge in the DR. Furthermore, Bohm and Stone's [27] method takes component names as input, derives their function structures based on the knowledge in the DR, and suggests alternative components. While all these methods demonstrate the potential to facilitate concept generation and exploration, their usefulness is inherently tied to the existing knowledge stored in the design repository.

While function is central to design knowledge, other types of knowledge are also essential in a design process. Fenves et al. [15] hence developed Core Product Model, a modeling language that aims to cover a more comprehensive set of design information, such as requirement, specifications, function, behavior, geometry, material, etc. of artifacts. While the model seems promising, we are not aware of any application of this model. Li and Ramani [16] also developed a comprehensive ontology that includes information such as products, components, materials, manufacturing, and environment. The main purpose of their ontology is to facilitate design information retrieval.

Acquisition of design knowledge

Li et al. [33] applied their previously developed ontology [16], along with natural language processing techniques, to retrieve formalized design knowledge from design reports. The important characteristic of their approach is that it uses specific knowledge of known entities and functions, e.g., using the Functional Basis terms, to support information retrieval. Our approach is similar in that we leverage Functional Basis to classify function knowledge. However, we also focus on extraction of unknown function definitions that might not involve the Functional Basis terms.

Zeng's [34] method uses a formal logical language called the recursive object model to represent objects and their relationships in a generic model. The author applied the method to automatically translate requirements documents into UML diagrams. Colombo et al. [35]'s work also takes an ontological approach to design knowledge modeling, focusing on achieving a high level of formalism in describing relationships in mechanical products. For both of the efforts, the focus is more on extracting generic yet formal syntax of objects and relationships, e.g., a constraint between two objects, and less on identifying the specific semantics of objects or relations, such as known mechanical functions.

There are also various data mining efforts in design. Most related to our work, several researchers worked with text data to either retrieve useful information or classify text. Much of the work in the domain deals with the patent knowledge. For example, Cascini et al. [38] used syntactic parsing to identify

[3] http://design.engr.oregonstate.edu/repo

subject-action-object triplets in patent documents, which is the similar approach that our research follows. Other work such as Verhaegen et al. [39], Li et al. [40] and Murphy et al. [41] focus on classifying documents according to their usefulness in providing analogical inspiration or inventive principles. For biologically inspired design, Shu [36] has worked on applying natural language processing techniques to assist designers in finding relevant biological analogies to their design problems. The latest work from the author's group [37] used syntactic parsing and rules, much like our current work, to identify causally related functions in biology text. In addition, identifying customer needs from text resources have been recently explored, e.g., [42,43]. For most of the work described in this paragraph, in contrast to the knowledge acquisition work described in the previous paragraph, the information acquired is not formalized. Our goal is not only to extract design knowledge, but also work toward automatic construction of a formal knowledge base.

Word classification methods

An important challenge of automatic knowledge extraction is classifying the retrieved knowledge to an established ontology or taxonomy. Because our retrieved knowledge is described in natural language text, we aim to leverage word classification methods to classify the retrieved knowledge.

Two main approaches can be used to classify words. The lexicon-based approach uses existing lexical knowledge bases such as WordNet [28], which organizes English words by their synonyms, hypernyms, hyponyms, etc. in a hierarchy. Several methods have been developed to compute semantic similarities between a pair of words based on their relative locations within the WordNet hierarchy. The best performing method according to Budanitsky and Hirst [44] is the Jiang-Contrath measure [45], which considers the information content of words calculated from a particular corpus as well as the locations of words in the WordNet hierarchy. The method can achieve 85% correlation with human raters in assigning semantic similarity between a pair of words [44]. The limitation of this approach is its reliance on a specific lexical knowledge base; hence, the accuracy of similarity measures is dependent on how lexicons are categorized in the database.

Another approach is corpus-based. This approach is based on statistical information found in a corpus without any background knowledge. The common method in this approach involves constructing a term-document matrix and transforming the matrix into a vector space, such as Latent Semantic Analysis (LSA) [46]. The method enables both the terms and documents to be treated as vectors in a common vector space. Hence, cosine similarity between word vectors could be computed to estimate their lexical similarity. Google has released the word2vec tool that leveraged a neural network language model to *learn* vector representations of words from a large corpus [29]. The authors report interesting operations that could be done with the learned representations of words and resembles a form of logical reasoning, e.g., "Paris – France + Italy = Rome" [29]. One major limitation of the corpus-based approach is that the features that define a particular word vector are latent and therefore it is hard to reason why certain words are related to each other. Also, the type of relations this approach can identify is limited to lexical similarity, unlike the WordNet-based approach where you could identify hypernym/hyponym relations between words.

METHOD

We present our method to acquire subject-verb-object (SVO) triplets from text using syntactic rules and classify the triplets as artifact-function-energy flow knowledge.

Retrieval of subject-verb-object triplets

We first outline selection, preparation, and parsing of text. Then we describe the storage of the parsed information and the syntactic rules used to acquire SVO triplets. Figure 1 depicts the process. The overall process is highly automated. Only the syntactic rules used to find triplets need to be manually defined once. The rest of the individual processes such as preparing the text, parsing the text, and acquiring triplets are fully automated.

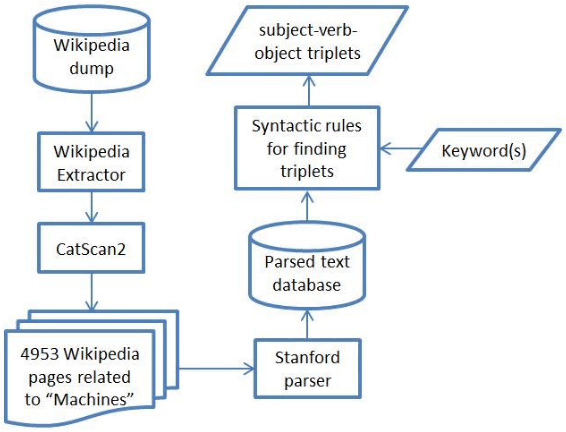
Wikipedia dump
Wikipedia Extractor
CatScan2
4953 Wikipedia pages related to "Machines"
Stanford parser
Parsed text database
Syntactic rules for finding triplets
Keyword(s)
subject-verb-object triplets

Figure 1: Subject-verb-object triplet acquisition process

Selection and preparation of corpus

We chose Wikipedia as the corpus for our current research. The choice was made for the breadth of the topics it contains and its availability as open, digital data. Other prior research in automated knowledge extraction has also used Wikipedia as the source corpus [10-12]. However, our method is not limited to a specific corpus, but can work with any English natural language text written in complete sentences. We downloaded the Wikipedia database dump[4] available on June 12, 2014. We then used Wikipedia Extractor[5] to keep only the text content from the database dump.

While we processed and parsed the entire Wikipedia contents, the current research focused on knowledge extraction from a small subset of Wikipedia pages related to mechanical engineering objects. We used the CatScan2 tool[6] to identify all

[4] http://en.wikipedia.org/wiki/Wikipedia:Database_download

[5] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

[6] http://tools.wmflabs.org/catscan2/catscan2.php

pages that are classified in the category “Machines”, its direct subcategories, and their direct subcategories. For example, the page “Bolt (fastener)” will be included as our test corpus because it is classified under the category “Fasteners”, which is a subcategory of “Hardware (mechanical)”, which is a subcategory of “Machines”. Based on this criterion, we identified 4953 Wikipedia pages as our test corpus.

Parsing of text

Prior to parsing, we processed the entire text to insert a period at each end-of-line that does not end with a period. This was because the parser used periods to determine the end of a sentence or phrase.

The Stanford parser [47] was used to parse the text. For each sentence taken as input, the parser outputs Penn Treebank part-of-speech (POS) tags [48] for the words and typed dependencies [49] between pairs of words. Typed dependencies are grammatical relations between two words. For example, in the sentence “The shaft transmits torque”, the words “shaft” and “transmits” are in a relation called “nominal subject”. These typed dependencies are the key information that we use to identify SVO triplets. The Stanford parser was chosen particularly because it produces these typed dependencies. Table 1 shows an example output of the parser. All the parsed information was stored in a PostgresSQL database for the efficient retrieval of triplets.

Syntactic rules for finding subject-verb-object triplets

Table 2 lists the rules used to identify SVO triplets based on syntactic information in a single sentence. Three combinations of typed dependencies are used: nominal subject (nsubj) + direct object (dobj), controlling subject (xsubj) + direct object (dobj), and agent (agent) + passive nominal subject (nsubjpass). The first case identifies sentences with direction association between a subject and a verb, e.g., “The shaft transmits torque”, while the second case identifies sentences with indirect association between a subject and a verb, e.g., “The shaft is used to transmit torque.” The third case identifies association between a subject and a verb in sentences with a passive voice, e.g., “Torque is transmitted by the shaft”. In all cases, we also identify the relationship between the verb and the direct object. In addition, noun compound modifier (nn) and adjectival modifier (amod) are used to identify compound nouns, e.g., nn(shaft, drive) ⇒ “drive shaft” and adjective + noun phrases, e.g., amod(shaft, rear) ⇒ “rear shaft”.

Retrieval of subject-verb-object triplets

Our method can take any combination of subject, verb, and object as keywords. For example, if the goal was to identify a function of a particular artifact, a subject keyword such as “shaft” could be used. This scenario is the focus of our current research. If the goal was to identify a list of artifacts based on a function-flow definition, a verb-object keyword combination such as “transmit-torque” could be used. Our search tool also features searching for different forms of the verb and nouns.

Table 1: Example parser output

Example sentence:
“The rear shaft transmits torque.”

Part-of-speech tags:
The/DT rear/JJ shaft/NN transmits/VBZ torque/NN ./.
DT: Determiner
JJ: Adjective
NN: Noun, singular or mass
VBZ: Verb, 3rd person singular present

Typed dependencies:

det(shaft-3, The-1)	*determiner*
amod(shaft-3, rear-2)	*adjectival modifier*
nsubj(transmits-4, shaft-3)	*nominal subject*
dobj(transmits-4, torque -5)	*direct object*

Table 2: Rules to find subject-verb-object triplets

For all cases,[7]
POS(x, VB*) ∧ POS(y, NN*) ∧ POS(z, NN*)
[VB, VBD, VBG, VBN, VBP, VBZ] ⊆ VB*
[NN, NNS, NNP, NNPS] ⊆ NN*

1. nsubj(x, y) ∧ dobj(x, z)
⇒ Subject-Verb-Object(y, x, z) ∧ Subject(y) ∧ Verb(x) ∧ Object(z)
Example: “The shaft transmits torque”

2. xsubj(x, y) ∧ dobj(x, z)
⇒ Subject-Verb-Object(y, x, z) ∧ Subject(y) ∧ Verb(x) ∧ Object(z)
Example: “The shaft is used to transmit torque”

3. agent(x, y) ∧ nsubjpass(x, z)
⇒ Subject-Verb-Object(y, x, z) ∧ Subject(y) ∧ Verb(x) ∧ Object(z)
Example: “Torque is transmitted by the shaft”

Classification of triplets as artifact-function-flow

We classify the SVO triplets retrieved as artifact-function-flow knowledge using the combination of the Functional Basis terms, WordNet, and word2vec. For the current work, we focus on classifying only energy flows, but the approach could be applied to classify other types of flows such as material and signal.

Classification of functions

To classify verbs in SVO triplets as function terms, we used the Functional Basis function set as the reference. In Functional Basis, functions are classified into eight primary classes, and 21 secondary classes. For the current work, we focus on classifying knowledge at the secondary class level, because it is recommended for and most commonly used in function modeling [24,50].

Each secondary function class was populated with function keywords in the following manner. First, for each secondary class, we included the name of the class, the names of tertiary functions in the class, and all the correspondents as function terms. Then, for each individual term, we used WordNet to find

[7] NN* and VB* indicate different forms of nouns and verbs

its synonyms. In WordNet, each sense of a word is grouped with other words that have the same meaning, as a "synset". We took the members of the synset in which the appropriate sense of each function term was found. Using this approach, we were able to identify 401 unique function keywords in total, classified amongst 21 secondary function classes. As an example, Table 3 shows the list of keywords included in the function class, "couple".

Table 3: Example of keywords included in the secondary function class, "Couple"

Second class	Tertiary class	Correspondents	WordNet synsets
couple	join, link	associate, connect, assemble, fasten, attach	conjoin, fix, piece, put together, secure, set up, tack, tie

Because this approach results in a large number of function keywords, we classified the verbs of SVO triplets based on simple matching to our function keywords. In future, we plan to investigate including more keywords using the troponyms, i.e., more specific forms of a verb, or developing a classifier similar to the method that we use to classify flows (described in the next section).

Some of the keywords are repeated in different classes. This would result in a particular verb being classified in more than one function class. We decided to keep every classification because we value recall over precision. In general, our approach aims to capture the available knowledge in text at the expense of incorrectly classifying some of the retrieved knowledge.

Classification of energy flows

In contrast to the function set in Functional Basis, the flows do not contain as many correspondents. In addition, most of the secondary and tertiary class names are adjectives used to modify the primary class names. For example, "acoustic energy" is a secondary flow class under the primary class, "energy". Hence, it is difficult to populate a comprehensive set of keywords for each flow class using WordNet and use keyword-based classification for flows.

Therefore, we developed a classification method that estimates whether particular noun phrases are energy flows; and if so, the method classifies the energy flows as one of 11 secondary energy classes in Functional Basis (we ignored the class "Human Energy", because it is essentially "Mechanical Energy" delivered by a human). The energy classes are listed in Table 4. We need to classify noun phrases, e.g., "electrical power", not just the head noun "power", because the leading adjectives or noun modifiers provide strong cues for classifying objects in SVO triplets (which could be a noun or a noun phrase) according to the secondary energy class.

Table 4: Secondary energy classes in Functional Basis

Human	Acoustic	Biological	Chemical
Electrical	Electromagnetic	Hydraulic	Magnetic
Mechanical	Pneumatic	Radioactive/Nuclear	Thermal

The reason to focus on classifying energy flows is as follows. First, the majority of function definitions (52.2% according to Caldwell et al. [50]) in the DR consist of energy flows. For signal flows, they represent information that is carried by corresponding material or energy flows [31]. Similarly, text descriptions involving signal flows are often implicitly expressed using material or energy flows, e.g., "a LED emits light." Because inferring implicit meanings is difficult, we only focus on classifying such descriptions as knowledge involving material or energy flows. We are also still working on a method to classify objects as material flows at the secondary class levels in Functional Basis. However, the general approach for the material flow classification method should be similar to the energy flow classification method explained below.

Figure 2 shows the overall approach of the energy flow classification method. At the first step, if the object of a SVO triplet matches one of stand-alone power conjugates defined for an energy flow class [24], the method classifies the object as the corresponding flow class. For example, "torque" is a stand-alone power conjugate for "Mechanical Energy". Ten stand-alone power conjugates are defined in Functional Basis, providing only a limited set of keywords for classification.

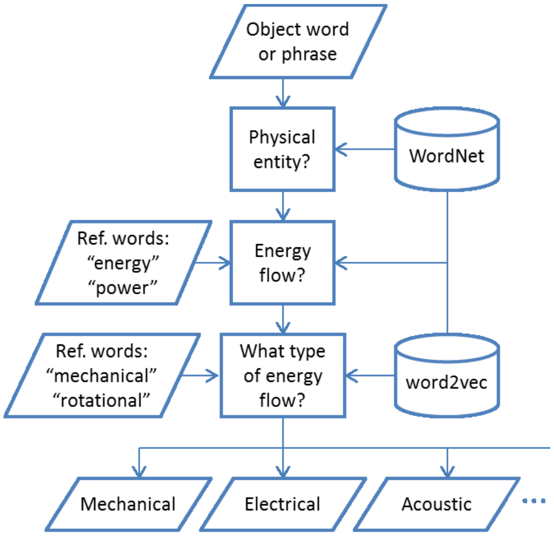

Figure 2: The energy flow classification method

If the object was not one of stand-alone power conjugates, the method checks to see if all the possible senses of the object noun (or the head noun if the object is a noun phrase) are classified under "abstract entity" in the WordNet hierarchy. In such a case, the triplet containing the object is not considered as useful. For example, in the description, "The experiment supports the theory", "theory" would be identified as an abstract entity, and therefore the triplet would be discarded.

Next, the method uses the following function to identify the likelihood of the object being an energy flow:

$$u = \text{sim}(\overrightarrow{W}_{energy}, \overrightarrow{W}_{object}) + \max_{s_x} \text{jcn}(s_{energy}, s_x) \qquad (1)$$

The function takes the sum of two similarity measures found using word2vec and WordNet.

The sim() function finds the cosine similarity between two word vectors, $\overrightarrow{W}_{energy}$ and $\overrightarrow{W}_{object}$, obtained using word2vec. The word2vec tool provides a vector of n dimensions for a given word. The word vector is defined in a vector space model trained on a chosen corpus. For our work, we use the pre-trained model provided by Google based the Google News corpus[8]. $\overrightarrow{W}_{energy}$ is defined as:

$$\overrightarrow{W}_{energy} = word2vec(\text{"energy"}) + word2vec(\text{"power"}) \quad (2)$$

The words "energy" and "power" are chosen because they characterize the nature of energy flows used in Functional Basis. $\overrightarrow{W}_{object}$ is defined as:

$$\overrightarrow{W}_{object} = \sum_{w \in P} word2vec(w) \quad (3)$$

where w indicates individual words of the noun phrase, P, identified as the object. If the object is a single noun, only that noun word is used.

Finally, jcn() is the Jiang-Conrath similarity measure [45] between a pair of word senses, s, found in WordNet. s_{energy} is the first sense of "energy" defined in WordNet. Because we do not know the sense of the object to be classified, we compute the Jiang-Conrath similarity measure between s_{energy} and every sense of the object, s_x, and use the maximum similarity measure found.

We used the threshold of $u > 2.9$ to determine if the object is an energy flow or not. This threshold value maximizes the accuracy (F-measure) in classifying the test terms listed in Table 5. These test terms consist of examples given by Hirtz et al. [24] for each flow type and additional energy flows that we brainstormed. The chosen threshold value achieved the F-measure of 0.82, with 75% precision and 90% recall.

Once the object has been identified as an energy flow, we used word2vec and cosine similarity measures to further classify the object according to the secondary energy classes defined in Functional Basis. This classification can be expressed as:

$$\underset{x \in E}{\operatorname{argmax}} \ \operatorname{sim}(\overrightarrow{W}_x, \overrightarrow{W}_{object}) \quad (4)$$

where $\overrightarrow{W}_x$ is the sum of word vectors for all the class names associated with the secondary energy class, with $x \in E$; $E = \{electrical, mechanical, acoustic, \dots\}$. For example, for the energy class "Electromagnetic Energy",

$$\begin{aligned} &\overrightarrow{W}_{electromagnetic} \\ &= word2vec(\text{"electromagnetic"}) \\ &+ word2vec(\text{"energy"}) + \text{word2vec("optical")} \\ &+ word2vec(\text{"solar"}) \end{aligned} \quad (5)$$

The first two words form the name of the secondary energy class, while the latter two words come from the names of its children classes. Our method achieved 90% accuracy in correctly classifying the test terms listed in Table 5 according to their associated energy flow classes.

Table 5: Test terms used for the energy flow classification

Material flows	**Energy flows**	*Energy flow class*
air	beam	*Electromagnetic*
water	sound	*Acoustic*
sandpaper	sound waves	*Acoustic*
box	microwaves	*Electromagnetic*
granular sugar	bond	*Chemical*
powdered paint	voltage	*Electrical*
wood	light	*Electromagnetic*
fiberglass	electricity	*Electrical*
Kevlar cloth	strain	*Mechanical*
rain	stress	*Mechanical*
snow	fluid power	*Hydraulic/Pneumatic*
sleet	solar power	*Electromagnetic*
oil	Lorentz force	*Magnetic/Electrical*
gasoline	EMF	*Electrical/Electromagnetic*
concrete	infrared radiation	*Electromagnetic*
plaster	ionizing radiation	*Radioactive/Nuclear*
iced tea	nuclear fission	*Radioactive/Nuclear*
fog	entropy	*Thermal*
soda	ultraviolet	*Electromagnetic*
ice cubes	friction	*Mechanical*
smoke		
mist		
blood		
fluid		
aerosol		

Classification of artifacts

For the current work, we only check whether all senses of the subject in SVO triplets are under "abstract entity" in WordNet. Taxonomy of electromechanical components, as proposed by Kurtoglu et al. [51], would facilitate classification of artifacts. The same approach for classifying flows, e.g., using a set of reference component names, WordNet, and word2vec, could be applied to classify unknown component names.

EVALUATION

We first evaluate the precision of SVO triplets retrieved. Then, we compare the artifact-function-energy flow knowledge classified by our method to the knowledge compiled in the DR.

Precision of triplets retrieved

We evaluated the precision of our method in terms of retrieving factually correct SVO triplets. We used the function terms of Functional Basis, as well as their synonyms, as verb keywords to search for triplets from our test corpus.

Out of 4653 total triplets found, 500 triplets were randomly sampled for evaluation. Three independent raters assigned a true or false rating to each of the 500 triplets. For each triplet, the raters were also provided with the sentence in which the triplet was found. Hence, the raters evaluated whether the SVO triplet is a true proposition found in the sentence.

On average, 89% of the triplets retrieved were factually correct, indicating high precision. The raters had an average pairwise agreement of 93.9%, with Fleiss's Kappa of 0.67, which indicates substantial agreement [52].

[8] https://code.google.com/p/word2vec/

Comparison to knowledge in Design Repository

After evaluating the precision of SVO triplets, we evaluated how well our method can classify the triplets as artifact-function-energy flow knowledge. For this evaluation, we compared the knowledge extracted using our method to those human-compiled in the DR.

We identified 30 test artifacts from the DR based on the frequency of function definitions assigned to each artifact. For the analysis, we grouped artifacts that have the same head noun in their names, e.g., the function definitions for “wire”, “black wire”, and “wire 2” would be tallied together. Table 6 shows the list of the top 30 artifacts selected.

We used the names of 30 artifacts as subject keywords and extracted function-energy flow definitions from our test corpus. Our method was able to identify 557 function-energy flow definitions for 24 artifacts, shown in Table 6. In general, the method retrieved function knowledge for most of the machinery components. This is likely due to the choice of our test corpus, which focused on Wikipedia pages classified under the category “Machines”. With an expanded corpus including more diverse topics, we should be able to find function knowledge for the missing artifacts. For the rest of the evaluation discussion below, we focus on the artifacts with their function knowledge identified.

We computed the proportion of the function-energy flow definitions assigned to each test artifact in the DR that our extraction method also identified. For this comparison, we ignored function definitions involving material, signal, and human energy flows, as we neglected classification of these flows for our current research. We also translated all function definitions with tertiary energy flows, e.g. “rotational energy”, into secondary energy flows, e.g., “mechanical energy.” Finally, we only considered input flows in the function definitions of artifacts. For example, for the artifact “motor”, given input flow = “electrical energy”, function = “convert”, and output flow = “mechanical energy”, we formed “convert-electrical energy” as the function definition for the artifact. In most cases, input flows and output flows are identical in the DR.

Figure 3 shows the proportion of function definitions assigned to each test artifact in the DR that our method was able to identify. For an artifact such as “gear”, the coverage is over 90%. However, for artifacts such as “plug” and “housing”, the coverage was minimal. Overall, our method identified 50% of the function definitions in the DR.

Another measure of accuracy is to examine how much of the function definitions identified by our method was compiled in the DR. This indicates how much of the function knowledge extracted could be irrelevant. Figure 4 shows that comparing function definitions at the secondary class level, 27% of the function definitions identified by our method were also found in the DR. This indicates that our method tends to identify many false positives. However, because the function definitions compiled in the DR is not guaranteed to be complete or correct, our method may have identified new function definitions for the test artifacts that did not exist in the DR.

Table 6: Thirty most frequent artifacts in Design Repository

Function knowledge found by our method		Function knowledge not found by our method
wire	wheel	circuit board
gear	blade	cord
spring	engine	cover
motor	rotor	bowl
switch	bearing	handle
shaft	cable	heating element
tube	screw	support
battery	impeller	axle
plug	guide	reservoir
housing	trigger	solder

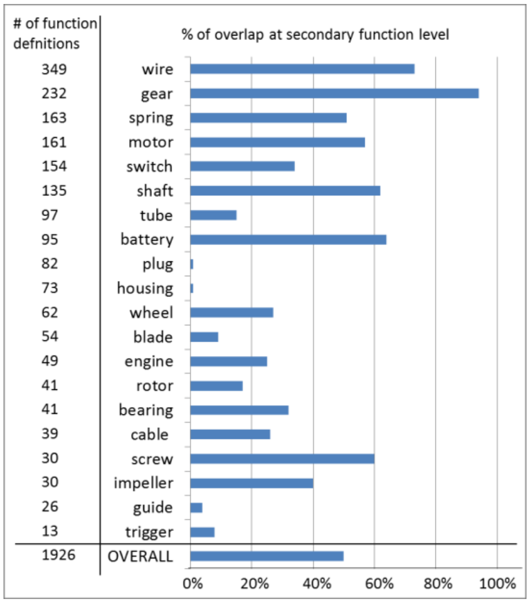

Figure 3: Proportion of function definitions assigned to test artifacts in Design Repository, identified by our method

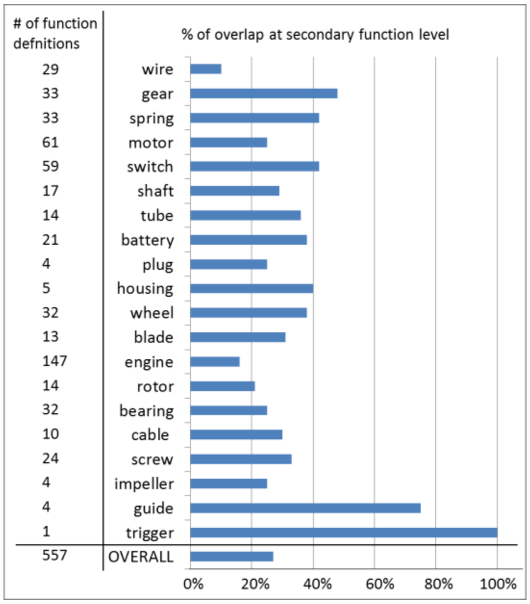

Figure 4: Proportion of function definitions identified by our method for test artifacts, assigned in Design Repository

Finally, Table 7 lists the most frequent function definitions that our method identified for each test artifact. For 75% of the artifacts, the most frequent function definitions identified was also compiled in the DR. While our method made incorrect classification for “wire”, “shaft”, and “wheel” (e.g., “guide-hydraulic” identified for “wheel”), for “engine” and “bearing”, one could argue that the function definitions identified, “supply-mechanical” and “regulate-mechanical”, respectively, are correct but did not exist in the DR.

Table 7: Most frequent function-energy flow definitions extracted by our method for test artifacts.

Artifact name	Function definition (secondary)	Frequency	Defined in DR?
wire	change-chemical	3	N
gear	transfer-mechanical	6	Y
spring	supply-mechanical	6	Y
motor	supply-mechanical	9	Y
switch	guide-electrical	3	Y
shaft	transfer-electrical	2	N
tube	transfer-thermal	2	Y
battery	supply-electrical	6	Y
plug	supply-electrical	1	Y
housing	stop-electrical	1	Y
wheel	guide-hydraulic	4	N
blade	supply-mechanical	2	Y
engine	supply-mechanical	8	N
rotor	convert-electrical	2	Y
bearing	regulate-mechanical	5	N
cable	transfer-electrical	2	Y
screw	import-mechanical	2	Y
impeller	convert-mechanical	1	Y
guide	regulate-mechanical	2	Y
trigger	convert-mechanical	1	Y
	% of functions defined in DR:		**75%**

DISCUSSION

The evaluation results demonstrate the viability of our method in automatically extracting artifact-function-energy flow knowledge from text. Using sentence-level parsing, syntactic rules, and concept classification methods, we are able to extract specific knowledge that can be compiled in a formal knowledge base. This approach contrasts from conducting document-level analysis to classify or cluster text. Knowledge compiled in a formal manner, such as the DR, can be used for computational concept generation methods [25-27] and various reasoning tasks.

Our method leveraged a combination of a lexicon-based approach (WordNet) and a vector space model (word2vec) to classify energy flow terms used in engineering design. This approach resolved the issue of the sparse vocabulary in the Functional Basis flow taxonomy [50]. In general, we used WordNet for high-level concept classification and word2vec for low-level concept classification. The wod2vec tool worked very well if specific and representative terms for concept categories were used as reference word vectors (e.g., 90% accuracy for the secondary energy classification in our test). However, if a general term such as “energy” was used, it showed limitations. For instance, the word “oil” would be found as highly similar to the word “energy”, because of their frequent co-occurrences in a typical corpus. Hence, we used WordNet, a lexical knowledge base, to mitigate this limitation.

An important benefit of our method is that statistics of function definitions can be obtained. Such statistics could assist researchers or designers in resolving ambiguities that occur during function modeling. In addition, the statistics could be used in deciding which components to recommend to designers given a function definition. For example, the synthesis rule set developed by Kurtoglu and Campbell [26] replaces a give function definition with corresponding artifacts. The choice of replacement rules could be made based on the statistical relationships between function definitions and artifacts observed in a corpus.

While our current work focuses on extracting function knowledge, the extraction method could be used to identify other types of design knowledge as well. For instance, the SVO triplets used in our work is similar to the subject-action-object triplets used by Cascini et al. [38] to construct semantic networks from a patent document. In addition, Li et al. [33] used a similar approach based on syntactic parsing to identify relationships amongst parts, materials, and manufacturing processes found in a design report. The SVO triplets obtained in our current work retrieved information such as “swingarm-holds-rear axle”, “motor-has-commutator”, or “stamping-processes-metal”, all of which could be useful design knowledge. The main challenge would be finding appropriate methods to classify such knowledge.

We believe that further formalization of the Functional Basis taxonomy, such as the work done by Sen et al. [53], would tremendously assist the knowledge extraction efforts. For instance, the function-flow pair of “support-thermal energy” likely does not have valid semantics; hence, for SVO triplets that contain “support” as their verbs, we can eliminate the possibility of classifying the objects as thermal energy. In addition, many of the tertiary-level function terms can be used in the same context in text, e.g., “increase” vs. “increment”. Perhaps one could introduce ontological axioms to formally distinguish the meanings of these terms based on the flows that they can be paired with. For example, an axiom could define that “increase” could only be used with “materials” and “increment” could only be used for “energy” or “signal”.

CONCLUSIONS AND FUTURE WORK

So far, our work focused on identifying function-energy flow definitions for given artifacts from 4953 Wikipedia pages. The method was able to identify 50% of the function definitions assigned to selected test artifacts in Oregon State University’s Design Repository (DR). In addition, 75% of the most frequent function definitions that our method identified for those artifacts were present in the DR. Even with the limited size of the test corpus, our method achieved a fair bit of coverage in extracting relevant function knowledge compared to the DR, a human-compiled knowledge base that has existed for over 15 years.

We can take several research directions to expand on this work. First, we plan to develop classification methods for other types of flows, i.e., material and signal. We also need more sophisticated methods for classifying artifacts, or even non-physical entities such as manufacturing processes or material properties. We need further experimentation with WordNet, word2vec, and other potentially useful taxonomies, such as NASA QUDT [54]. In addition, we are looking into applying bootstrapping techniques with a list of known artifact or flow names to improve our classification methods.

By expanding our corpus, we could gather statistics and patterns of function knowledge. Self-supervision methods used in machine learning, such as redundancy and joint inference [13], could be used with the statistical knowledge to further improve our extraction method. For example, by considering only redundant knowledge, our method's tendency to identify many false positives, as indicated in Figure 4, could be reduced. Also, we could enable aggregation of new knowledge without the need to validate against a "labeled" knowledge base such as the DR. Another approach could be to focus on more domain-specific and published corpora, such as mechanical engineering handbooks or patent documents. While Wikipedia served as an accessible, large-scale source for the current work, its quality could be less reliable and precise than published sources. In addition, training the word2vec tool on a more domain-specific corpus would likely improve our flow classification method.

Another important challenge is that we are only extracting isolated knowledge for each artifact. Ideally, we would like to obtain system-level knowledge such as "electrical energy → engine-convert → mechanical energy → shaft-transfer → mechanical energy". This requires distinguishing input and output flows for each artifact, and reference resolutions to extract associated knowledge from multiple sentences. Our eventual goal is to extract formalized function models of mechanical systems from text. We believe that the current work lays an important step toward the goal.

ACKNOWLEDGMENTS

We thank Design Engineering Lab at Oregon State University for sharing the Design Repository data.

REFERENCES

[1] Zeng, Y., & Horváth, I. (2012). Fundamentals of next generation CAD/E systems. Computer-Aided Design, 44(10), 875-878.

[2] Gero, J. S. (1985). Knowledge Engineering in Computer-Aided Design: Proceedings of the IFIP WG 5.2 Working Conference on Knowledge Engineering in Computer-Aided Design. Elsevier Science Inc.

[3] Chandrasegaran, S. K., Ramani, K., Sriram, R. D., Horváth, I., Bernard, A., Harik, R. F., & Gao, W. (2013). The evolution, challenges, and future of knowledge representation in product design systems. Computer-Aided Design, 45(2), 204-228.

[4] Rocca, G. L. (2012). Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. Advanced engineering informatics, 26(2), 159-179.

[5] Verhagen, W. J., Bermell-Garcia, P., van Dijk, R. E., & Curran, R. (2012). A critical review of knowledge-based engineering: An identification of research challenges. Advanced Engineering Informatics, 26(1), 5-15.

[6] Tomiyama, T. (2007). Intelligent computer-aided design systems: Past 20 years and future 20 years. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 21(01), 27-29.

[7] Agichtein, E., & Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In the Fifth ACM Conference on Digital Libraries (pp. 85-94). ACM.

[8] Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., & Shadbolt, N. R. (2003). Automatic ontology-based knowledge extraction from web documents. Intelligent Systems, IEEE, 18(1), 14-21.

[9] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction for the web. In IJCAI, 7, 2670-2676.

[10] Auer, S. & Lehmann, J. (2007). What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content. In 4th European Semantic Web Conference

[11] Suchanek, F., Kasneci, G. & Weikum, G. (2007). YAGO - A core of semantic knowledge. In WWW 2007.

[12] Wu, F., & Weld, D. S. (2010). Open information extraction using Wikipedia. In the 48th Annual Meeting of the Association for Computational Linguistics (pp. 118-127).

[13] Poon, H., & Domingos, P. (2010). Machine reading: A "Killer App" for statistical relational AI. In Statistical Relational Artificial Intelligence.

[14] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E. R., & Mitchell, T. M. (2010). Toward an Architecture for Never-Ending Language Learning. In AAAI, 5, 3-11.

[15] Fenves, S. J., Foufou, S., Bock, C., & Sriram, R. D. (2008). CPM2: A core model for product data. Journal of Computing and Information Science in Engineering, 8(1), 014501.

[16] Li, Z., & Ramani, K. (2007). Ontology-based design information extraction and retrieval. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 21(02), 137-154.

[17] Ullman, D.G. (1992). The Mechanical Design Process. New York: McGraw–Hill.

[18] Otto, K.N., & Wood, K.L. (2001). Product Design Techniques in Reverse Engineering and New Product Development. Upper Saddle River, NJ: Prentice Hall.

[19] Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.H. (2007). Engineering Design: A Systematic Approach, 3rd ed. London: Springer-Verlag.

[20] Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design. AI Magazine, 11(4), 26.

[21] Umeda, Y., Takeda, H., Tomiyama, T., & Yoshikawa, H. (1990). Function, behaviour, and structure. Applications of artificial intelligence in engineering V, 1, 177-194.

[22] Chandrasekaran, B., Goel, A. K., & Iwasaki, Y. (1993). Functional representation as design rationale. Computer, 26(1), 48-56.

[23] Szykman, S., Sriram, R. D., Bochenek, C., & Racz, J. (1999). The NIST design repository project. In Advances in Soft Computing (pp. 5-19). London: Springer.

[24] Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., & Wood, K. L. (2002). A functional basis for engineering design: reconciling and evolving previous efforts. Research in Engineering Design, 13(2), 65-82.

[25] Bryant, C. R., McAdams, D. A., Stone, R. B., Kurtoglu, T., & Campbell, M. I. (2005). A computational technique for concept generation. In Proc. of ASME 2005 IDETC/CIE.

[26] Kurtoglu, T., & Campbell, M. I. (2009). Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping. Journal of Engineering Design, 20(1), 83-104.

[27] Bohm, M. R., & Stone, R. B. (2010). Form Follows Form: Fine tuning artificial intelligence methods. In Proc. of ASME 2010 IDETC/CIE.

[28] Miller, G. A. (1995). WordNet: A Lexical Database for English. Communications of the ACM, 38(11), 39-41.

[29] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:1301.3781.

[30] Goel, A. K. (1997). Design, analogy, and creativity. IEEE expert, 12(3), 62-70.

[31] Stone, R. B., & Wood, K. L. (2000). Development of a functional basis for design. Journal of Mechanical Design, 122(4), 359-370.

[32] Bohm, M., Stone, R., Simpson, S., & Steva, L. (2006). Introduction of a data schema: The inner workings of a design repository. In Proc. of ASME 2006 IDETC/CIE.

[33] Li, Z., Yang, M. C., & Ramani, K. (2009). A methodology for engineering ontology acquisition and validation. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 23(1), 37-51.

[34] Zeng, Y. (2008). Recursive object model (ROM) - Modelling of linguistic information in engineering design. Computers in Industry, 59(6), 612-625.

[35] Colombo, G., Mosca, A., & Sartori, F. (2007). Towards the design of intelligent CAD systems: An ontological approach. Advanced Engineering Informatics, 21(2), 153-168.

[36] Shu, L. H. (2010). A natural-language approach to biomimetic design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 24(4), 507-519.

[37] Cheong, H., & Shu, L. H. (2014). Retrieving causally related functions from natural-language text for biomimetic design. Journal of Mechanical Design, 136(8), 081008.

[38] Cascini, G., Fantechi, A., & Spinicci, E. (2004). Natural language processing of patents and technical documentation. Document Analysis Systems VI (pp. 508-520). Berlin: Springer.

[39] Verhaegen, P. A., D'hondt, J., Vandevenne, D., Dewulf, S., & Duflou, J. R. (2011). Identifying candidates for design-by-analogy. Computers in Industry, 62(4), 446-459.

[40] Li, Z., Tate, D., Lane, C., & Adams, C. (2012). A framework for automatic TRIZ level of invention estimation of patents using natural language processing, knowledge-transfer and patent citation metrics. Computer-Aided Design, 44(10), 987-1010.

[41] Murphy, J., Fu, K., Otto, K., Yang, M., Jensen, D., & Wood, K. (2014). Facilitating design-by-analogy: Development of a complete functional vocabulary and functional vector approach to analogical search. In Proc. of ASME 2014 IDETC/CIE.

[42] Rai, R. (2012, August). Identifying key product attributes and their importance levels from online customer reviews. In Proc. of ASME 2012 IDETC/CIE.

[43] Stone T., & Choi, S-K. (2014). Visualization tool for interpreting user needs from user-generated content via text mining and classification. In Proc. of ASME 2014 IDETC/CIE.

[44] Budanitsky, A., & Hirst, G. (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. Workshop on WordNet and Other Lexical Resources (Vol. 2).

[45] Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In 10th International Conference on Research in Computational Linguistics, ROCLING'97.

[46] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. JASIS, 41(6), 391-407.

[47] Toutanova, K., & Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Volume 13 (pp. 63-70). ACL.

[48] Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. Computational linguistics, 19(2), 313-330.

[49] De Marneffe, M. C., MacCartney, B., & Manning, C. D. (2006, May). Generating typed dependency parses from phrase structure parses. In LREC (Vol. 6, pp. 449-454).

[50] Caldwell, B. W., Sen, C., Mocko, G. M., & Summers, J. D. (2011). An empirical study of the expressiveness of the functional basis. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 25(03), 273-287.

[51] Kurtoglu, T., Campbell, M. I., Bryant, C. R., Stone, R. B., & McAdams, D. A. (2005). Deriving a component basis for computational functional synthesis. In ICED 05.

[52] Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. Biometrics, 33, 159-174.

[53] Sen, C., Summers, J. D., & Mocko, G. M. (2011). A protocol to formalise function verbs to support conservation-based model checking. Journal of Engineering Design, 22(11-12), 765-788.

[54] Hodgson, R., Keller, P. J., Hodges, J., & Spivak, J. (2014) QUDT – Quantities, Units, Dimensions and Data Types Ontologies. Retrieved from http://www.qudt.org/.