Proceedings of the ASME 2016 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2016
August 21-24, 2016, Charlotte, NC, USA

DRAFT of IDETC2016-59551

AUTOMATED EXTRACTION OF SYSTEM STRUCTURE KNOWLEDGE FROM TEXT

Hyunmin Cheong, Wei Li, Francesco Iorio
Autodesk Research
Toronto, ON, Canada

ABSTRACT

This paper presents a method to automatically extract structure knowledge of mechanical systems from natural language text. The current work extends our prior work on extracting function knowledge from text, which was presented at last year's conference. The method uses rules based on a combination of syntactic, lexical, and redundancy information to identify structure knowledge from parsed text. Three case studies were conducted to evaluate the method. The case studies involved extracting physical connections among a known set of components of a bicycle frame, an internal combustion engine, and a drum brake from Wikipedia. The current work makes progress toward addressing the challenge of knowledge acquisition for knowledge-based CAD systems.

INTRODUCTION

Knowledge is central in ascribing intelligence to any computational system. For a computer-aided design (CAD) system to support and perform intelligent tasks such as problem formulation and design synthesis, it inevitably requires an extensive design knowledge base [1-4]. However, knowledge acquisition is still a significant challenge in constructing a comprehensive and practical knowledge base for design.

To facilitate the knowledge acquisition process for knowledge-based CAD systems, our previous work presented a method to automatically extract function knowledge from text [5]. The method applied techniques of automated knowledge base construction, in particular machine reading [6], to extract "artifact-function-flow" knowledge (e.g., "gears-transfer-mechanical energy") from natural language text. Such function knowledge is essential in reasoning with design problems and solutions expressed at an abstract level to support conceptual and creative synthesis tasks.

The current work extends the function knowledge extraction method to acquire system structure knowledge from text. Here, system structure knowledge is defined as physical connections between components in a system. One prominent example of such knowledge is the internal block diagram represented in SysML [7]. An internal block diagram consists of blocks, which are structural elements of a (sub-) system, and their interconnections, as shown in Figure 1. Similarly, the function modeling diagrams [8], which are widely used in engineering, also convey system structure knowledge. The diagrams represent relationships between components as input / output flows, which can be material, energy, or signal [9].

System structure knowledge can be useful in several aspects of problem formulation during a computer-aided design process [2]. First, generalized system structure knowledge can serve as a template (or a "prototype" [10]) for new design problems. For example, a designer working on a bicycle frame design could import a generic system model such as Figure 1 and instantiate, refine, or modify the model according to the designer's specific problem. In addition, formally expressed structure knowledge combined with the function knowledge of individual components could produce a behavioral model of the system that can be simulated and optimized. This corresponds to formulating a system-level design problem. Or, the system structure knowledge can lead to the formulation of part-level design problems, e.g., shape optimization of a single or multiple part(s) in the system, as illustrated in Figure 2.

The method developed for the current work uses the similar approach as our work on function knowledge extraction [5]. First, a corpus is chosen and parsed to obtain syntactic information. Then, extraction rules are applied to identify candidate knowledge. The rules leverage both the syntactic information produced by the parser and the lexical knowledge of Functional Basis [9], WordNet [11] and word2vec [12].

The new contributions of the current work are the following. First, rules that are specific to extracting system structure knowledge are presented. In addition, the current work acquires graph data to represent system-level knowledge, in contrast from triplet data acquired in [5] to represent part-level knowledge. The knowledge acquired is also generalized from a larger corpus based on redundancy information – repeated observation of same knowledge from different documents [13].

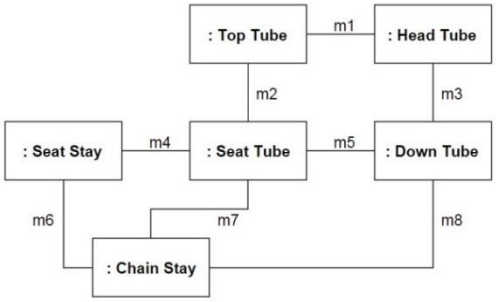

Figure 1: Example of an internal block diagram of bicycle frames, with edges representing mechanical connections

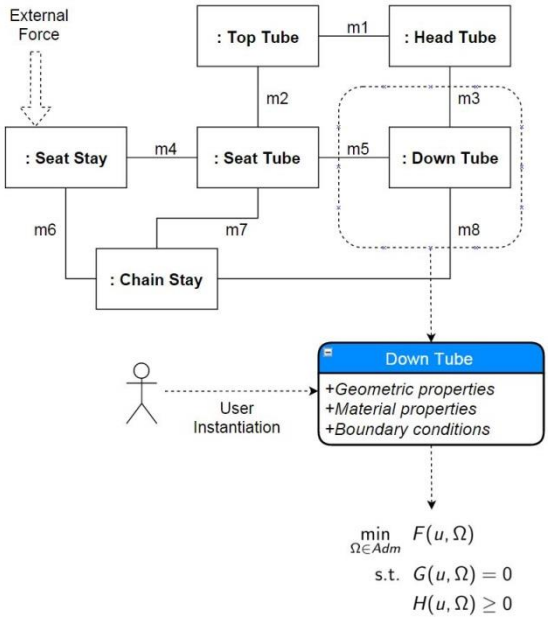

Figure 2: Illustration of formulating a part design optimization problem from system structure knowledge

The rest of the paper is outlined as follows. First, prior work in design knowledge modeling is highlighted, followed by general techniques used in knowledge acquisition from text and their applications to design knowledge acquisition. Then, the extraction method developed is presented, followed by the case studies conducted to evaluate the performance of the method. The paper ends with discussion, future work, and conclusions.

BACKGROUND

Design knowledge representation and reasoning

Much of the knowledge representation work for design has been rooted in function modeling [8]. Prominent knowledge representation frameworks include Gero et al. [10], Umeda et al. [14], Chandrasekaaran et al. [15], and Chakrabarti and Bligh [16]. Many of these frameworks use a graphical model to express system knowledge. This is because engineering design inherently deals with systems of parts. Even at an abstract level of representation such as function models, system knowledge must be expressed.

While many modeling frameworks have been established, very few large-scale design knowledge bases have been developed and used. To the best of our knowledge, the most extensive knowledge base is the design repository maintained by Oregon State University[1], originally initiated by NIST [17]. The repository contains function models of 184 products and 6906 artifacts. Based on this knowledge base, several synthesis methods have been developed [18]-[20]. All these synthesis methods apply reasoning on the graph-based representation of function models. Interestingly, these methods leverage the knowledge compiled in the repository. Hence, the usefulness of the methods depends on the amount of knowledge acquired.

Techniques for knowledge acquisition from text

Acquiring knowledge from text requires the analysis of semantics. To analyze semantics, the computer must identify relations between linguistic elements in text, e.g., words, and determine the meaning of those elements.

Two main approaches are used to identify useful relations in text. First, the syntactic analysis approach uses parsers to determine grammatical relations between words in a sentence. Traditionally, various chart parsing techniques, which use dynamic programming to disambiguate potential grammar combinations in a sentence, were used for syntactic analysis. Recently, the performance of statistical parsers has surpassed chart parsers [21]. Statistical parsers determine the most likely grammatical relations in a sentence based on the probability distribution obtained from data. The current work used Stanford Parser [22], a statistical parser that outputs typed dependencies as the form of grammatical relations for a given sentence. The main limitation of this approach is that grammatical relations are found only within a single sentence at a time.

Another approach used to uncover relations in text is distributional semantics. The essence of this approach is to capture and represent the co-occurrence of linguistic elements, e.g., words, over large corpora. Typically, vectors and matrices are used to capture the co-occurrence information, and techniques such as singular value decomposition can be used to find the most useful representation. Then, vector algebraic operations can be performed, such as calculating cosine similarity between document vectors. Latent semantic analysis is one popular method that embodies these techniques [23]. In addition, machine learning algorithms such as support vector machines can be used to classify document vectors. The main strength of this approach is its consideration of contextual information found in large sets of data. For the goal of determining semantic similarity, the approach works very well. However, it is typically limited to determining implicit relations, namely the semantic similarity.

As for analyzing lexical semantics, two distinct approaches are also used. The lexicon-based approach uses a lexical knowledge base created by domain experts. One example of a

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

lexical knowledge base is WordNet [11], created by linguists to capture the general semantics of English words. WordNet is hierarchical categorization of words by their lexical relations, such as synonyms, hypernyms, and hyponyms. Various methods have been developed to use the WordNet hierarchy to compute semantic similarity between a pair of words [24].

The distributional semantics approach described above can also be used to compute similarity between words. Hence, it is also useful in determining lexical semantics. One successful application of this approach for lexical semantics is word2vec developed by Google [12].

Our previous work [5] used a combination of Functional Basis [9], WordNet, and word2vec for lexical semantic disambiguation. Functional Basis [9] is a controlled set of function and flow terms used for function modeling. The current work also uses the same combination of lexical knowledge sources in the extraction method.

Knowledge acquisition from text for design

Both approaches of syntactic analysis and distributional semantics have been applied to extract design knowledge from text, depending on the purpose. To acquire explicit domain knowledge from a smaller corpus, the syntactic analysis approach is typically used. To acquire implicit knowledge across documents, the distributional semantics approach is used.

Li and Ramani [25] used syntactic parsing and their domain-specific ontology (a form of a lexical knowledge base) to extract concept graphs from design documents. Their concept graphs represent artifacts and their functional relationships. Interestingly, their subsequent work [26] focused on constructing a design ontology from engineering documents using syntactic and lexical analyses. Cascini et al. [27] also applied syntactic parsing to generate functional diagrams from patent documents. Their approach was unique and effective because the authors leveraged the reference numbers labeled on patent documents. Hence, the approach avoided the need to use a lexical knowledge base to disambiguate the entities described in documents. In biomimetic design, Cheong and Shu [28] used syntactic rules to retrieve causally related functions from biology texts, to support analogical reasoning. Recently, Kang et al. [29] attempted to extract manufacturing rules in the form of Semantic Application Design Language from text and Renu and Mocko [30] investigated potentially extracting assembly planning knowledge from assembly work instructions. While all of the above mentioned work used off-the-shelf parsers, Zeng [31] used the author's own recursive object model to translate requirements text into UML diagrams.

Much of the design research work using the distributional semantics approach focused on identifying analogical similarity between documents, which can be interpreted as the degree of two documents describing similar functional principles. For instance, Verhaegen et al. [32] and Vandevenne et al. [33] constructed term-document matrices and applied principal component analysis to identify analogous patent documents or biology texts, respectively. Murphy et al. [34] also used the term-document matrices as the main representation to identify functionally similar patent documents. All these authors discovered analogical similarity, instead of simple semantic similarity, by mainly lexical filtering. Verhaegen et al. used WordNet to filter noun artifact words, Vandevenne et al. filtered organism names in biological texts, and Murphy et al. filtered all noun terms and kept only the relevant words that belong to their function vocabulary. Other applications of distributional semantics include the use of latent semantic analysis to determine knowledge convergence in design teams [35] and a probabilistic approach to classify manufacturing suppliers [36]. In addition, Tuarob and Tucker [37] used a graphical network as a novel method of capturing co-occurrence information to extract implicit customer preferences from Twitter data.

The current work aims to extract and generalize system-level knowledge obtained from a large corpus. This contrasts from the work of Li and Ramani [25], Cascini [27], and Zeng [31], which also aimed to extract system-level knowledge, but focusing on a single document at a time. Also, while the distributional semantics approach is typically used to extract knowledge from a larger corpus, the current work applies the syntactic analysis approach. This allows obtaining explicit knowledge that can be interpreted by both humans and computers, eventually used in knowledge-based CAD systems.

METHOD

This section defines the current knowledge extraction problem, followed by the general approach used to solve the problem. Then, the extraction method is presented – parsing of a corpus, extraction and redundancy rules used to acquire relevant knowledge, and the implementation details.

Problem

The problem involves extracting system structure knowledge from text in the following form. Here, the system structure knowledge is defined as an undirected graph $G = (V, E)$, of which vertices (V) represent components of in a system and edges (E) represent physical connections between each pair of vertices, if present. For the current work, the set of vertices are assumed to be known (i.e., the components that belong to the system of interest are known), and the goal is to determine the edges (physical connections) based on the evidence obtained from unstructured text data. Hence, the problem can be stated as:

$$Given\ V = \{V_1, V_2, \dots, V_n\}$$
$$Find\ E_{i,j}(V_i, V_j);\ for\ i \le n, j \le n, i \ne j$$

General approach

The general approach to the knowledge extraction method can be defined as the following:

1. Select a corpus.
2. Parse the corpus to identify syntactic relations found in each sentence.

3. Apply the extraction rules, which use a combination of syntactic and lexical information, to obtain relevant evidence from the parsed corpus.
4. Apply the redundancy rule to determine the knowledge sought from the evidence obtained.

Figure 3 summarizes the knowledge extraction method developed. The details of each step are described in the following subsections.

Selection and parsing of a corpus

The current research goal is to obtain generalized knowledge over a large data set. Hence, Wikipedia was chosen as the corpus because of its size (2.9 billion words[2]) and the breadth of the topics it contains. In addition, it is available as open and digital data. More domain specific corpus such as patent descriptions of electromechanical products could be used in the future. The extraction method developed is not limited to a specific corpus, but can work with any English corpus written in complete sentences.

A Wikipedia database dump[3] was downloaded on June 12, 2014. Wikipedia Extractor[4] was used to extract only the text content from the database dump. The entire Wikipedia text was used as the corpus for knowledge extraction because the goal was to extract generalized knowledge across multiple documents.

The text was preprocessed by inserting a period at each end-of-line that does not end with a period, such as titles, section headers, and list items. This allows the parser used, Stanford Parser [22], to determine the end of a sentence or a phrase. The parser takes each sentence in the text as input and outputs Penn Treebank part-of-speech (POS) tags [38] for the words and typed dependencies [38] between pairs of words. Table 1 shows an example of parser output. Typed dependencies are grammatical relations between two words in a sentence. For example, in the sentence "The bracket holds the shelf against the wall", the words "bracket" and "holds" are in a relation called "nominal subject", indicating that the "bracket" is the subject of the verb, "holds". Stanford Parser was chosen mainly because it produces these typed dependencies. All the parsed information was stored in a PostgreSQL[5] relational database for efficient retrieval during the extraction process.

Knowledge extraction rules

This section first describes the assumptions made on the types of syntactic and lexical information found in sentences that convey the physical connection knowledge. Discussed next is the challenge of component names expressed in compound nouns, and how the extraction method handles them. Formal descriptions of the extraction rules are presented and explained at the end of the section.

[2] https://en.wikipedia.org/wiki/Wikipedia:Size_comparisons
[3] http://en.wikipedia.org/wiki/Wikipedia:Database_download
[4] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor
[5] http://www.postgresql.org/

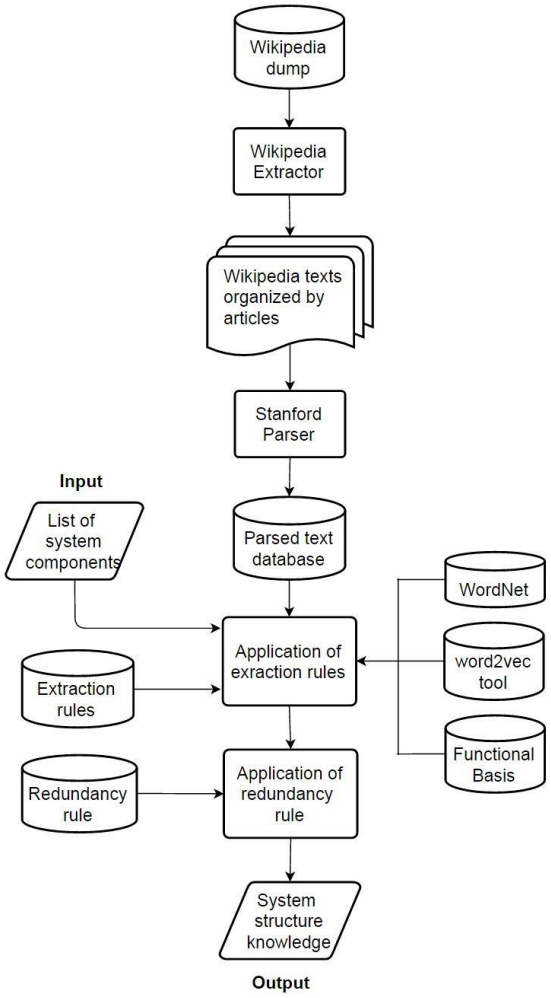

Figure 3: Summary of the knowledge extraction method

Table 1: Example parser output

Example sentence:
"The top tube connects the head tube to the seat tube."

Part-of-speech tags:
The/DT top/JJ tube/NN connects/VBZ the/DT head/NN tube/NN to/TO the/DT seat/NN tube/NN ./.

DT: *Determiner*
JJ: *Adjective*
NN: *Noun, singular or mass*
VBZ: *Verb, 3rd person singular present*
TO: *Infinitival to*

Typed dependencies:

det(tube-3, The-1)	*determiner*
amod(tube-3, top-2)	*adjectival modifier*
nsubj(connects-4, tube-3)	*nominal subject*
det(tube-7, the-5)	*direct object*
nn(tube-7, head-6)	*noun compound modifier*
dobj(connects-4, tube-7)	*direct object*
prep_to(connects-4, tube-11)	*prepositional modifier, to*
det(tube-11, the-9)	*determiner*
nn(tube-11, seat-10)	*noun compound modifier*

We identified two scenarios when the physical connection knowledge is conveyed in sentences. The first scenario is when a sentence contains a verb that conveys the meaning of a physical connection and is grammatically related with the components of interest. For example, in the sentence "The bracket holds the shelf against the wall", a physical connection between "bracket" and "shelf" is inferred if the verb "holds" is assumed to convey the meaning of a physical connection and participates in grammatical relations with the nouns "bracket" and "shelf", and the two nouns are the components of interest. In summary, the following deduction is desired:

Given:
Extraction Rule A
+
"The bracket **holds** the shelf against the wall"

PhysicalConnection("bracket", shelf"),
PhysicalConnection("bracket", "wall")

The following technique is used to determine whether a verb found in a sentence conveys a physical connection. First, we manually selected a set of verbs from Functional Basis that convey physical connections. All of these verbs came from two primary classes in Functional Basis, namely "Connect" and "Support". The set of verbs selected are {"connect", "join", "assemble", "fasten", "link", "attach", "support", "stabilize", "steady", "secure", "hold", "fix"}.

This set of selected verbs is likely not all the verbs in the English language that convey physical connections. To increase the likelihood of identifying a physical connection verb from sentences, WordNet [11] was used to expand the selected verb set. For each verb in the original set, the *synset* of the verb was identified. In WordNet, each word is classified into synsets based on one of the word's multiple possible meanings. Hence, synsets can be thought as classes of synonyms. Figure 4 shows the synsets identified for each function verb and their hierarchical relationships found in WordNet.

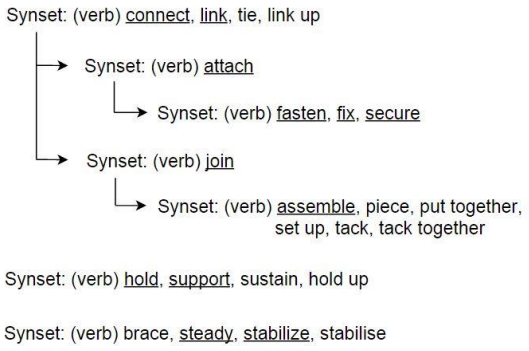

Figure 4: Selected verbs from Functional Basis (underlined) and their synsets found in the WordNet hierarchy ("is-a" relation)

After locating the appropriate synset for each word, the original set of selected verbs was expanded to include the synonyms and all the troponyms that belong under the synset. Troponyms are more specific forms of a verb, e.g., "fasten" is a troponym of "attach". We call this expanded set of verbs as *P*. Hence, if a verb found in a sentence belongs to this set *P*, the verb is assumed to convey a physical connection.

The second scenario of the physical connection knowledge expressed in a sentence is assumed to be the following. A sentence contains a noun that is one of specific flows [9] conveying a physical connection and participates as the direct object of a verb, and that verb is grammatically related with the components of interest. For example, in the sentence "The piston transfers the force to the crankshaft", the noun "force" is a mechanical energy flow that is the direct object of the verb "transfers", which is grammatically related with the nouns "piston" and "crankshaft". Therefore, a physical connection between "piston" and "crankshaft" is assumed. In summary, the following deduction operation is desired:

Given:
Extraction Rule B
+
"The piston transfers **force** to the crankshaft"

PhysicalConnection("piston", crankshaft")

The physical connection knowledge is assumed only when specific types of energy flows are involved. First, a material or signal flow is neglected because either flow could be transferred between components without a physical connection between them (e.g., via the medium of air). Among the energy flow types, a physical connection is assumed when electrical, mechanical, hydraulic, or pneumatic energy flows are involved. Flow types such as acoustic or electromagnetic energy can be transferred across objects without a physical connection.

To determine whether a particular noun is a flow type of our interest, the flow classification method developed in [5] is applied, which demonstrated 90% accuracy. The method involves first classifying whether a noun is an energy flow using a combination of WordNet-based and word2vec-based similarity measures. Then, it further classifies the noun into a specific energy flow class using the word2vec tool. For the details of the method, please refer to [5]. Using this method, we can define *F* as a set of nouns that are classified as the flows conveying physical connections.

We now describe the technique to handle component names. In many mechanical systems, component names are expressed as compound nouns. Compound nouns consist of a head noun modified with adjectives or other nouns. For example, "top tube" is the head noun "tube" modified with the adjective "top", "head tube" is a the head noun "tube" modified with another noun "head", and "bottom bracket shell" is the head noun "shell" modified with the adjective "bottom" and another noun "bracket".

The following data structure is used to express input system components. It is a set of component names, where each component name is a set of up to three elements. An example for a bicycle system could be the following:

$$E^{bicycle} = \{\{\text{"fork"}\}, \{\text{"head"}, \text{"tube"}\}, \{\text{"bottom"}, \text{"bracket"}, \text{"shell"}\}, \dots\}$$

In theory, there is no limit on how many adjectives or nouns can be used to modify the head noun. However, for the current work, the capability of the knowledge extraction method is limited to handle only compound nouns consisting of up to three words, which is deemed sufficient for evaluating the feasibility of our approach. Handling compound nouns that consist of any number of words would require the program to dynamically generate the rules at the run time.

The extraction process iterates through all possible combinations of a pair of component names for a given system, and finds from text the evidence of a physical connection between each pair. Because the number of words allowed for a component name is limited to three, nine different cases of extraction rules are required. The nine cases represent the pair-wise combinations (3×3) of component names that can have one, two, or three words.

When collecting the evidence of a physical connection between a pair of components expressed as compound nouns, two different types of evidence are considered – complete and partial. Given a pair of compound nouns, e.g., "rear shaft" and "rear wheel", the complete evidence scenario corresponds to when the physical connection knowledge is observed between the full names of the compound nouns, e.g., "The rear axle transfers torque to the rear wheel." The partial evidence scenario corresponds to when the physical connection knowledge is observed between the head nouns of the compound nouns, e.g., "An axle transfers torque to a wheel." Both types of evidence are kept during the extraction process, while assigning different confidence weights to each type of evidence to conclude whether a physical connection actually exists between the given pair of components.

Table 2 and Table 3 present the formal descriptions of our knowledge extraction rules based on the knowledge of physical connection verbs and flows, respectively. The following paragraph on the next page explains the rules in detail.

Table 2: Extraction rules based on the knowledge of verbs that convey physical connections

For all cases:

$P = set\ of\ verbs\ conveying\ physical\ connection$
$R1 = nsubj(z, x_0) \cap (dobj(z, y_0) \cup pobj(z, y_0))$
$R2 = agent(z, x_0) \cap (nsubjpass(z, y_0) \cup pobj(z, y_0))$
$R3 = xsubj(z, x_0) \cap (dobj(z, y_0) \cup pobj(z, y_0))$

Case 1: Both component names consist of three-word compound nouns, $V_i = \{x_2, x_1, x_0\}$ and $V_j = \{y_2, y_1, y_0\}$.

$\forall x_0, x_1, x_2, y_0, y_1, y_2, z$
$P(z) \cap (R1 \cup R2 \cup R3)$
$\cap\, nn(x_0, x_1) \cap amod(x_0, x_2) \cap nn(y_0, y_1) \cap amod(y_0, y_2)$
$\rightarrow \boldsymbol{PhysConnect}(\{x_2, x_1, x_0\}, \{y_2, y_1, y_0\})$

$\forall x_0, y_0, z$
$\exists x_1, x_2, y_1, y_2$
$P(z) \cap (R1 \cup R2 \cup R3) \cap x_0 \neq y_0$
$\rightarrow \boldsymbol{PhysConnectPartial}(\{x_2, x_1, x_0\}, \{y_2, y_1, y_0\})$

Case 2: The first component name consists of three-word compound nouns, $V_i = \{x_2, x_1, x_0\}$, and the second component name consists of two-word compound nouns, $V_j = \{y_1, y_0\}$.

$\forall x_0, x_1, x_2, y_0, y_1, z$
$P(z) \cap (R1 \cup R2 \cup R3)$
$\cap\, nn(x_0, x_1) \cap amod(x_0, x_2) \cap (nn(y_0, y_1) \cup amod(y_0, y_1))$
$\rightarrow \boldsymbol{PhysConnect}(\{x_2, x_1, x_0\}, \{y_1, y_0\})$

$\forall x_0, y_0, z$
$\exists x_1, x_2, y_1$
$P(z) \cap (R1 \cup R2 \cup R3) \cap x_0 \neq y_0$
$\rightarrow \boldsymbol{PhysConnectPartial}(\{x_2, x_1, x_0\}, \{y_1, y_0\})$

…

Case 9: Both component names consist of single nouns, $V_i = \{x_0\}$ and $V_j = \{y_0\}$.

$\forall x_0, y_0, z$
$P(z) \cap (R1 \cup R2 \cup R3)$
$\rightarrow \boldsymbol{PhysConnect}(\{x_0\}, \{y_0\})$

Table 3: Extraction rules based on the knowledge of flows that convey physical connections

For all cases:

$F = set\ of\ flow\ nouns\ conveying\ physical\ connection$
$R1 = nsubj(z, x_0) \cap dobj(z, w) \cap (iobj(z, y_0) \cup pobj(z, y_0))$
$R2 = agent(z, x_0) \cap dobj(z, w) \cap (nsubjpass(z, y_0) \cup pobj(z, y_0))$
$R3 = xsubj(z, x_0) \cap dobj(z, w) \cap (dobj(z, y_0) \cup pobj(z, y_0))$

Case 1: Both component names consist of three-word compound nouns, $V_i = \{x_2, x_1, x_0\}$ and $V_j = \{y_2, y_1, y_0\}$.

$\forall x_0, x_1, x_2, y_0, y_1, y_2, w$
$\exists z$
$F(w) \cap (R1 \cup R2 \cup R3)$
$\cap\, nn(x_0, x_1) \cap amod(x_0, x_2) \cap nn(y_0, y_1) \cap amod(y_0, y_2)$
$\rightarrow \boldsymbol{PhysConnect}(\{x_2, x_1, x_0\}, \{y_2, y_1, y_0\})$

$\forall x_0, y_0, w$
$\exists x_1, x_2, y_1, y_2, z$
$F(w) \cap (R1 \cup R2 \cup R3) \cap x_0 \neq y_0$
$\rightarrow \boldsymbol{PhysConnectPartial}(\{x_2, x_1, x_0\}, \{y_2, y_1, y_0\})$

Case 2: The first component name consists of three-word compound nouns, $V_i = \{x_2, x_1, x_0\}$, and the second component name consists of two-word compound nouns, $V_j = \{y_1, y_0\}$.

$\forall x_0, x_1, x_2, y_0, y_1, w$
$\exists z$
$F(w) \cap (R1 \cup R2 \cup R3)$
$\cap\, nn(x_0, x_1) \cap amod(x_0, x_2) \cap (nn(y_0, y_1) \cup amod(y_0, y_1))$
$\rightarrow \boldsymbol{PhysConnect}(\{x_2, x_1, x_0\}, \{y_1, y_0\})$

$\forall x_0, y_0, w$
$\exists x_1, x_2, y_1, z$
$F(w) \cap (R1 \cup R2 \cup R3) \cap x_0 \neq y_0$
$\rightarrow \boldsymbol{PhysConnectPartial}(\{x_2, x_1, x_0\}, \{y_1, y_0\})$

…

Case 9: Both component names consist of single nouns, $V_i = \{x_0\}$ and $V_j = \{y_0\}$.

$\forall x_0, y_0, w$
$\exists z$
$F(w) \cap (R1 \cup R2 \cup R3)$
$\rightarrow \boldsymbol{PhysConnect}(\{x_0\}, \{y_0\})$

Conditions $P(z)$ in Table 2 and $F(w)$ in Table 3 correspond to whether a relevant verb or a flow noun is found in the sentence. Conditions *R1*, *R2*, and *R3* ensure that the relevant verb or the flow noun is associated with the component pairs of interest through appropriate grammatical relations. Different *R* conditions are required for different forms of sentences. *R1* is applicable to sentences in the active voice, e.g., "The bracket supports the shelf." *R2* is applicable to sentences in the passive voice, e.g., "The shelf is supported by the bracket." *R3* is applicable to sentences when the verb of interest follows an infinitive, e.g., "The shelf is used to support the bracket." As explained before, different cases of rules are required to handle different combinations of compound nouns used as component names. Here, only a few cases are presented due to the space limit. Finally, the evidence is collected as either *PhysConnect* or *PhysConnectPartial* based on whether the complete compound nouns or only the head nouns are involved in the observation. The grammatical relations *nn* and *amod*, which correspond to the "noun modifier-head noun" and "adjective-head noun" relationships, are used to identify compound nouns.

Redundancy rule

Applying the extraction rules collects evidence of the physical connection knowledge from text. The method then determines how much evidence is enough to conclude that a physical connection exists between a pair of components. To make this generalization, the concept of redundancy [13] is applied – the observed knowledge is assumed to be true if the knowledge is found multiple times across different documents. Also, because both the complete and partial evidence of the knowledge are acquired during our extraction process, different weights, w_c and w_p, can be applied to each evidence type, respectively. Hence, the following rule is applied to determine the physical connection knowledge, where $\{\boldsymbol{PhysConnect}(V_i, V_j)\}$ and $\{\boldsymbol{PhysConnectPartial}(V_i, V_j)\}$ are the sets of complete and partial evidence observed for the component pair of V_i and V_j.

$$E_{i,j}(V_i, V_j) \; exists \; if:$$

$$w_c|\{\boldsymbol{PhysConnect}(V_i, V_j)\}| + w_p|\{\boldsymbol{PhysConnectPartial}(V_i, V_j)\}| \geq u$$

The values for the parameters w_c, w_p, and u can be determined based on training data. The Case Studies section presents the best parameter values found for three different examples.

Implementation of the extraction algorithm

A program was written in Python to implement the knowledge extraction method. The program first accesses the parsed text database, applies the extraction and redundancy rules, and returns the system structure knowledge obtained for a given set of components. The program also leveraged two Python libraries – gensim[6] was used for its word2vec tool and NLTK (Natural Language Toolkit)[7] was used for WordNet-based computing.

[6] http://radimrehurek.com/gensim/

[7] http://www.nltk.org/

CASE STUDIES

Three case studies were conducted to evaluate the extraction method. We selected three archetypical examples of mechanical systems – a bicycle frame, an internal combustion engine, and a drum brake. These systems consist of a well-defined set of components that are physically connected. In addition, we could identify Wikipedia pages (bicycle frame[8], internal combustion engine[9], drum brake[10]) that list the typical components found in these systems. The three systems also had a similar number of components listed in those Wikpedia pages (n=8 for a bicycle frame, n=9 for an internal combustion engine, and n=8 for a drum brake).

After identifying the set of components for each system, we created the ground truth knowledge by manually defining physical connections between the components where they exist. This ground truth knowledge was established by reading the relevant materials in the Wikipedia pages and other sources found on the Internet. Figures 5, 7, and 9 depict the ground truth knowledge defined for the systems. The goal of our experiment was to compare the ground truth knowledge to the knowledge obtained with our extraction method.

The experiment tested different sets of redundancy rule parameters. Because all the parameters are relative measures, the value of w_c, the weight assigned for the complete evidence, was held constant, while testing different values of w_p, the weight assigned for the partial evidence, and u, the threshold value used to determine the physical connection knowledge. The values of w_p were chosen to be smaller than the values of w_c, because w_p is the weight for the weaker form of evidence. Hence, we set w_c = 1, the domain of w_p as [0, 0.1, 0.2, 0.3, 0.4, 0.5], and the domain of u as [1, 2, 3, 4, 5]. Hence, 30 tests are performed in total for each case study.

The current problem consists of determining whether an edge exists or not for a given set of vertices. Hence, information retrieval measures such as precision and recall can be used to evaluate the accuracy of our method. Precision and recall are defined as the following:

$$precision = \frac{|\{relevant\ knowledge\} \cap \{retrieved\ knowledge\}|}{|\{retrieved\ knowledge\}|}$$

$$recall = \frac{|\{relevant\ knowledge\} \cap \{retrieved\ knowledge\}|}{|\{relevant\ knowledge\}|}$$

In other words, precision measures how much of the retrieved knowledge is true, while recall measures how much of the relevant knowledge of interest is found. F-measure combines the two measures by calculating its harmonic mean:

$$F = \frac{2 * precision\ * recall}{precision + recall}$$

[8] https://en.wikipedia.org/wiki/Bicycle_frame#Frame_tubes

[9] https://en.wikipedia.org/wiki/Component_parts_of_internal_combustion_engines#Parts

[10] https://en.wikipedia.org/wiki/Drum_brake#Components

Table 4: Accuracy results based on the best parameters

	Our Method						Baseline: word2vec			
	w_c	w_p	u	Precision	Recall	**F-measure**	v	Precision	Recall*	**F-measure**
Bicycle frame	1	0.1	1	1.00	0.778	**0.875**	0.1	0.375	*1.000*	**0.545**
Internal combustion engine	1	0.4	2	0.667	0.750	**0.706**	0.05	0.222	*1.000*	**0.364**
Drum brake	1	0.5	3	0.750	0.375	**0.500**	0.05	0.286	*1.000*	**0.444**

***The word2vec method with the best parameter, v, simply connected all the edges in the graph, resulting in 100% recall, but with low precision scores.*

Table 4 reports the accuracy measures calculated for the extraction method. Here, only the accuracy measures obtained for the best parameter set are reported. Ideally, one optimal set of parameters should be used to report the accuracy measures. However, we did not consider the three examples as a large enough data set to draw conclusions on the optimal parameter set. Finding the optimal parameter set based on more experiment data is planned as the future work. In addition, the sensitivity of the parameters should be analyzed to examine the robustness of the method. For now, we wanted to get a sense of the accuracy of the method based on the three case studies.

To get an idea of the difficulty of this extraction task, our method was compared to a baseline method. The baseline method used word2vec to find the vector representations of each component and determine the physical connection knowledge based on the cosine similarity between those vectors. For instance, the method concludes that there is a physical connection between two components if the cosine similarity is greater than some threshold value, v. Several experiments with this method were performed using different v values (domains: [0.05, 0.5], quantity: 10). Similar to reporting the results of our extraction method, the results obtained with the best parameter value were reported for the baseline method.

Figures 6, 8, and 10 depict the best results obtained with our method. These figures can be compared against the ground truth knowledge depicted in Figures 5, 7, and 9.

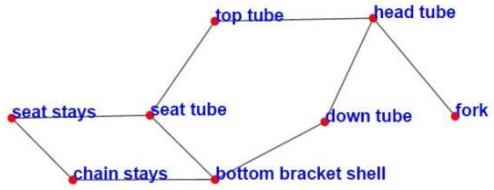

Figure 5: Ground truth knowledge for the bicycle example

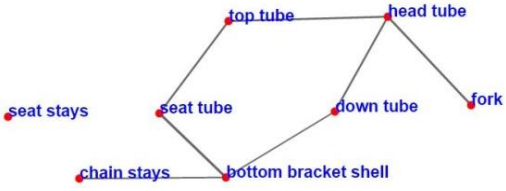

Figure 6: Knowledge obtained with the best redundancy parameters for the bicycle example

For the bicycle frame example, our method performed very well. The method obtained the F-measure of 87.5%, with 100% precision and 78% recall. For the internal combustion engine example, the method obtained fair results with the F-measure of 70%, precision of 67%, and recall of 75%. For the drum brake example, the method performed relatively poorly, with the F-measure of 50%, particularly because of the low recall score (38%), while having fair precision (75%).

For all three examples, our extraction method achieved higher F-measure scores than the baseline method. The best results for the baseline method essentially made all the possible physical connections between the components, obtaining 100% recall but low precision. Hence, it cannot be considered as reliable in distinguishing true knowledge from a false one. Our extraction method, on the other hand, tended to have higher precision scores (0.667-1.00) than recall scores (0.375-0.778).

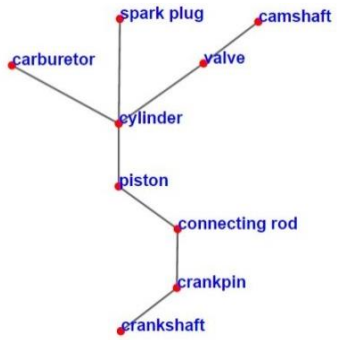

Figure 7: Ground truth knowledge for the engine example

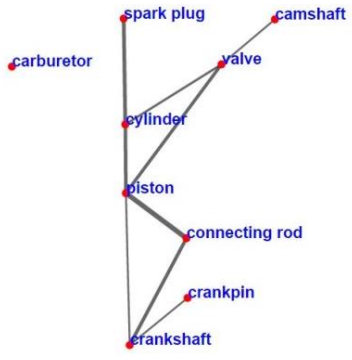

Figure 8: Knowledge obtained with the best redundancy parameters for the engine example

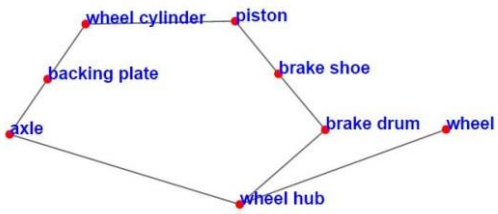

Figure 9: Ground truth knowledge for the drum brakes example

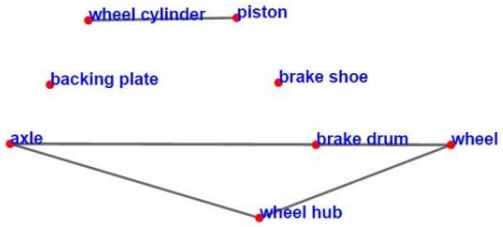

Figure 16: Knowledge obtained with the best redundancy parameters for the drum brakes example

DISCUSSION, FUTURE WORK, AND CONCLUSIONS

The main contribution of the current work is the extraction method developed to obtain generalized and explicit system-level knowledge across a large text data set. On the other hand, previous work on design knowledge acquisition from text focused on obtaining either explicit knowledge from a single text or implicit knowledge, e.g., analogical similarity, from a large corpus. The generalized knowledge obtained using our extraction method can be used as a template for systems modeling and design. In addition, because the knowledge obtained is explicit, it can be comprehended by both humans and computers for truth maintenance.

The current method could be combined with the method developed for function knowledge extraction [5] to acquire more semantically rich system models. Such models would not only contain the structure knowledge, but also the function knowledge of individual components. Acquiring the combination of both structure and function knowledge could enable automated construction of a design repository [17] that can take advantage of various computational synthesis methods [18]-[20].

The main limitation of the current method is that it can only extract knowledge from text if it is explicitly expressed in a single sentence. The relatively low recall scores reported in the case studies indicate this limitation. While humans can infer implicit semantics expressed across multiple sentences, it is still very challenging for computers to obtain such knowledge. However, the scalability of the extraction method could enable processing through even a larger data set and at least discover all the evidence of knowledge that is explicitly stated.

The current extraction method can be improved in several ways. First, as mentioned before, additional experiments should be conducted to obtain the optimal parameters for the redundancy rule. Next, we could investigate using different types of lexical knowledge in our extraction rules. For instance, a different set of verbs that more accurately indicate the physical connection knowledge could be identified. Ideally, the method should automatically identify not only the relations but also the component names of a given system. During the extraction, each component name could also be expanded with a class of synonymous component names to increase the chance of locating relevant knowledge. Naturally, a list of components for a given system or synonymous names for given components could be also acquired from text data. Finally, the text-based knowledge extraction method could be combined with other modes of knowledge acquisition methods such as diagram understanding [6].

The current work demonstrates the feasibility of extracting generalized system structure knowledge from text. We believe that the current work makes important progress toward addressing the long standing challenge of knowledge acquisition for enabling knowledge-based design systems.

ACKNOWLEDGMENTS

We thank Andy Nogueira and Ali Hashemi for their support in the software engineering work required for the current research.

REFERENCES

[1] 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.

[2] Coyne R. D., Rosenman, M. A., Radford, A. D., Balachandran, M., & Gero, J. S. (1990). Knowledge-based design systems. Addison-Wesley Publishing Company.

[3] 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.

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

[5] Cheong, H., Li, W., Cheung, A., Nogueira, A., & Iorio, F. (2015). Automatic extraction of function knowledge from text. Proceedings of ASME 2015 IDETC/CIE (DETC2015-47541), Boston, MA.

[6] Etzioni, O., Banko, M., & Cafarella, M. J. (2006). Machine Reading. In AAAI, 6, 1517-1519.

[7] Friedenthal, S., Moore, A., & Steiner, R. (2014). A Practical Guide to SysML: The Systems Modeling Language. Morgan Kaufmann.

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

[9] 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.

[10] Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design. AI Magazine, 11(4), 26.
[11] Miller, G. A. (1995). WordNet: A Lexical Database for English. Communications of the ACM, 38(11), 39-41.
[12] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:1301.3781.
[13] Downey, D., Etzioni, O., & Soderland, S. (2005). A probabilistic model of redundancy in information extraction. In Proceedings of the 19th international joint conference on Artificial intelligence (pp. 1034-1041). Morgan Kaufmann Publishers Inc.
[14] Umeda, Y., Takeda, H., Tomiyama, T., & Yoshikawa, H. (1990). Function, behaviour, and structure. Applications of artificial intelligence in engineering V, 1, 177-194.
[15] Chandrasekaran, B., Goel, A. K., & Iwasaki, Y. (1993). Functional representation as design rationale. Computer, 26(1), 48-56.
[16] Chakrabarti, A., & Blight, T. P. (2001). A schem for functional reasoning in conceptual design. Design Studies, 22, 493-517.
[17] 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.
[18] Bryant, C. R., McAdams, D. A., Stone, R. B., Kurtoglu, T., & Campbell, M. I. (2005). A computational technique for concept generation. Proceedings of ASME 2005 IDETC/CIE (DETC2005-85323), Long Beach, CA.
[19] 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.
[20] Bohm, M. R., & Stone, R. B. (2010). Form Follows Form: Fine tuning artificial intelligence methods. Proceedings of ASME 2010 IDETC/CIE (DETC2010-28774), Montreal, Quebec, Canada.
[21] Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1 (pp. 423-430). ACL.
[22] 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.
[23] 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.
[24] 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).
[25] Li, Z., & Ramani, K. (2007). Ontology-based design information extraction and retrieval. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 21(02), 137-154.
[26] 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.
[27] Cascini, G., Fantechi, A., & Spinicci, E. (2004). Natural language processing of patents and technical documentation. Document Analysis Systems VI (pp. 508-520). Berlin: Springer.
[28] Cheong, H., & Shu, L. H. (2014). Retrieving causally related functions from natural-language text for biomimetic design. Journal of Mechanical Design, 136(8), 081008.
[29] Kang, S, Patil, L., Rangarajan, A., Moitra, A., Jia, T., Robinson, D., Dutta, D. (2015). Extraction of Manufacturing Rules from Unstructured Text Using a Semantic Framework. Proceedings of ASME 2015 IDETC/CIE (DETC2015-47556), Boston, MA.
[30] Renu, R. S. & Mocko, G. (2015). Text Analysis of Assembly Work Instructions. Proceedings of ASME 2015 IDETC/CIE (DETC2015-47246), Boston, MA.
[31] Zeng, Y. (2008). Recursive object model (ROM) - Modelling of linguistic information in engineering design. Computers in Industry, 59(6), 612-625.
[32] 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.
[33] Vandevenne, D., Verhaegen, P. A., Dewulf, S., & Duflou, J. R. (2016). SEABIRD: Scalable search for systematic biologically inspired design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 30(01), 78-95.
[34] Murphy, J., Fu, K., Otto, K., Yang, M., Jensen, D., & Wood, K. (2014). Function based design-by-analogy: a functional vector approach to analogical search. Journal of Mechanical Design, 136(10), 101102.
[35] Dong, A. (2005). The latent semantic approach to studying design team communication. Design Studies, 26(5), 445-461.
[36] Yazdizadeh, P. Y., Ameri, F. A text mining technique for manufacturing supplier classification. Proceedings of ASME 2015 IDETC/CIE (DETC2015-46694), Boston, MA.
[37] Tuarob, S., & Tucker, C. S. (2015). A product feature inference model for mining implicit customer preferences within large scale social media networks. Proceedings of ASME 2015 IDETC/CIE (DETC2015-47225), Boston, MA.
[38] 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.
[39] De Marneffe, M. C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In LREC (Vol. 6, pp. 449-454).

