Exploring two possible means to identify issues in text

Madhusudanan N[1], B Gurumoorthy, Amaresh Chakrabarti

[1]Virtual Reality Lab,
Centre for Product Design and Manufacturing,
Indian Institute of Science, Bangalore, India
madhu@cpdm.iisc.ernet.in,
bgm@cpdm.iisc.ernet.in
ac123@cpdm.iisc.ernet.in

Abstract. The work reported in the paper is primarily aimed towards building a knowledge base for diagnosis of aircraft assembly procedures. The first step is to identify the presence of an issue in the text. Various existing methods that deal with issues in engineering are studied and their suitability is presented here. A study of sample documents across domains, including that of assembly is then presented. Some key observations from the study are discussed, following which two main methods are explored to detect issues. The first method is based on functional analysis of design, which deems that an issue is the result of the violation of a function or a related parameter. The second is the Natural Language Processing technique called Sentiment Analysis that aggregates sentiments from individual words. The suitability of these two methods is then discussed.

Keywords: Natural language text, diagnosis, sentiment analysis.

1 Introduction

Assembly is an important step in a product's lifecycle. It is an integrative step that sources inputs from the previous steps of design and part manufacturing. However a number of issues that affect assembly can be avoided at the design and planning stage itself using appropriate knowledge. Such knowledge of issues may be recorded in case studies, issue reports, change requests and other documents.

It is, however, a challenge for an individual, team or organization to meticulously search, read, and understand all such documents. There could be hundreds, possibly thousands of such documents that may be present within a large organization. These documents could be varied in nature, and may be spread across various domains, and across various parts of the product life-cycle.

The current research aims to capture the knowledge present in these documents offline, and present an assembly planner with such knowledge in a contextual manner. The general framework for this work is as shown in Figure 1.

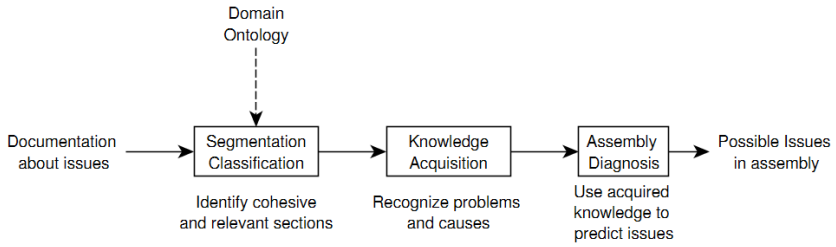

Fig. 1. A framework for acquiring diagnostic knowledge from documents.

1.1 Issues in product design and realization

There are issues that can be faced at any stage of a product's life-cycle. Examples of such stages include design [Hales], manufacturing, and usage. The current work being reported can be located in the assembly stage. Although generic guidelines have been proposed successfully for tackling such problems (e.g. DFA guidelines [Reference]), they may not cover some uncommon, but, important issues. These are issues that experienced personnel would have faced and documented.
The aircraft industry is the application area for this work. Due to the nature of the industry, assembly problems are not yet classified in a generic manner. One reason could be that it is still largely manual-assembly intensive, and hence a large number of cases of a many varieties are present, as opposed to assembly-lines, where a few classifications of problems is possible. Our aim is to make use of documents that contain instances of such issues, and acquire the required diagnostic knowledge.

2 Current methods:

Given that the objective of this paper is to identify methods for diagnosis, several methods in engineering diagnosis can be looked at. Identifying faults and issues, or their causes, has been studied for some time now. The following review only looks at methods to identify issues and causes rather than representing them, such as in the case of Ishikawa / Fishbone analysis.

A popular method applied in the industry is the Root Cause Analysis Method. It is largely a four-step process [Reference]: collecting data, drawing up a causal chart, identifying the root cause, and finally, suggesting a resolution based on the root-cause.

Another means of dealing with issues is the Finite Mode Effects and Analysis (FMEA) [Reference]. FMEA, by contrast, involves foreseeing the presence of various modes in which failure can occur. It is a managerial way of understanding a process, identifying possible failures in the process and addressing some of them. Once again, it is a largely manual means of organizing people and documents, resulting in improved processes with reduced failures.

The method of Fault-Tree Analysis (FTA) is perhaps closer to what we aim to achieve. FTA is a formal deductive method of identifying causes of an issue [FTA Handbook NASA reference]. Working backwards from the issue, a logical tree of the issue-cause relations is constructed. The final result is a probabilistic assessment of the likely causes and the chances of the issues occurring.

There has been previous work on modeling diagnostic knowledge in systems. Chandrasekaran et.al [Reference] looks at systems that can have two types of diagnosis. In particular they point to the required knowledge for one type, namely about malfunction hypotheses, and the relations between observations and malfunction hypotheses.

As far as natural language processing techniques are concerned, the most popular method till now is that of Sentiment Analysis. This objective for this set of methods is to find the overall sentiment of a piece of text by analyzing the individual sentiments. For example Taboada et al calculate a semantic orientation based on sentiments individual words, in combination with various modifying factors. Examples of such factors are valence shifters, such as intensifiers, downtoners and irrealis cases [Polanyi and Zaenen]. A general English network of words that indicate sentiment has also been in vogue for some time now [SentiWordNet reference]. This is a promising source of knowledge for realising whether a piece of text conveys a positive or negative sentiment. Commercial tools such as Lexalytics' Sentiment Analysis are also quite effective [Reference].

Given the constraint of not having a very large data set, and the fact that we are looking for going beyond just identification of issues, many of the above methods present difficulties to the current task at hand. FMEA looks at functions of each system, then possible failures for each - it would not help us since we have to first detect failures and then associate them with a system. Root Cause Analysis does post-analysis, inferring the root cause after a detailed understanding of the causal factors involved, and is usually a manual task. Regarding FTA, it is not possible to identify the presence of the issue from text using this method. The functional representation and sentiment analysis methods are more promising from the perspective of this work. They are explored in greater detail in the following sections.

Following a survey of existing methods, the need was felt to get a better understanding of means to identify issues in text. For this, it was necessary to understand how issues are represented in documents. In the next section we present a study in this regard.

3 Initial Study of Documents

The goal for this work is to acquire diagnostic knowledge to a level where it can be reused. To do so, the presence of issues in the text is important, as is the knowledge about the causes of the issues. This is shown in detail in Figure 2.

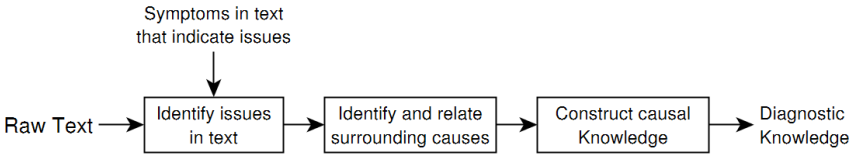

Fig. 2. Procedure to construct diagnostic knowledge.

In order to develop a means of acquiring diagnostic knowledge from documents, it is necessary to understand how such issues are recorded in text. For this, we surveyed a number of documents available openly, and extracted portions that contain text where reporting of these issues happen. A total of 20 such samples were studied. These samples were from different domains, ranging from consumer complaints to bicycle maintenance and aircraft riveting.

During the course of such a study, the exact words (or phrases) which indicated the presence of an issue or an undesirable state were manually marked. Where possible, the words (or phrases) that indicated one or more causes of the issue were also marked. The next step was to identify if any common pattern can emerge from such a study.

The following are some of the observations from the study:

- There are certain domain related key functions or parameters whose satisfaction or occurrence respectively (or negation) is seen as a problem
  e.g. "*application has been declined*" for a credit card domain;
  "*brakes do NOT work properly*" for a bicycle chain domain;
  "*rivet was too hard*" for a riveting domain;
  "*urinary chromium concentrations measured during BM-II were still higher than references from non-occupationally exposed populations*" for a workplace safety domain;
- There need not be a singular problem
  e.g. "*chain will slip and skip*";
  "*can cause slipping and may wear out drive train components*"
- There can be a linear chain of causes; it could be one problem that leads to another
  e.g. "*rivet head cracked because rivet was too hard when driven*"
  "*front left wheel assembly suddenly collapsed" LEADS TO "the driver immediately lost all steering control*"
- Usually, a singular or small set of root causes is not found, unless a large number of cases are analyzed
- Amount of background knowledge required to understand the utterance of an issue is large. Unlike human understanding, it cannot be assumed that the proposed system will have enough knowledge to implicitly understand the issue.
  e.g. "*known carcinogen to humans*" - unless the word carcinogen carries a negative value of sentiment, it is unlikely to be understood that this is a problem (because cancer is eventually caused).
- There are only three possible orders for causal reasoning - either build up, or post-analysis or both. The causes of the issue may be explained building up the final

issue, or the issue can be mentioned first followed by an explanation. Sometimes it can be a mix of the two.

- It is not yet known if the set of root causes is known or unknown - the condition for us is that they have to be identifiable with the system

The potential use of these observations is that we can construct possible templates for detecting issues in text, and breaking down the text into necessary pieces.

4 Detecting issues in text

Following our analysis of documents that contain issues, our next step is to identify what methods can be employed, and whether they can be used as is, or need to be modified for our purposes. Before this, it helps to clarify a few points about the planned method for implementation.

The need for diagnostic knowledge dictates need to acquire knowledge of issues as well as the knowledge about the causes of issues. Current methods do not seem to cover this portion about causes, unless it is about co-existence of the same terms across many documents. The need for understanding the document, as well as the subsequent use of logical form to do so using the Discourse Analysis method has been shown previously. Hence all such analysis of issues and causes in this work would be eventually performed on the logical form of text rather than the text itself. One example of a logical form (as given by the DRS tool Boxer [Reference]) is as follows. For the sentence

"*An aircraft has many parts.*"

the corresponding logical form is

patient(x1,x2), agent(x1,x3),have(x1), parts.(x2), quantity(x2), aircraft(x3);

where *x1, x2, x3* are the discourse entities.

From the discussion presented in the previous section, we foresee two possible methods using which presence of issues in text can be identified. The first method is related to the use of functions as a means of identifying negative text segments. The second method can utilize the vast corpus of work from the domain of Sentiment Analysis.

4.1 Domain functions

As discussed in the section on the study of documents, the presence of a domain related function or parameter can be clue to detecting an issue in text. The first of our proposed methods is to utilize this knowledge.

Functions of a product have been previously modeled using functional models [Wood / Chandrasekaran]. They may be useful for detecting undesired behavior of a system by means of negation of the function or any of its parameters. Literature exists about representing such functions [Chandrasekaran and Josephson, 2000], and building a basis of such functions for design [Stone and Wood, 2000]. Possible

sources of function-related information include functional ontologies, such as Design Repository [UM-Rolla reference].

Compiling such ontologies and resources is labor-intensive on its own. Hence, unless there are large-scale, domain relevant functional ontologies (that we are yet to access) it is not feasible to use this means to detect issues.

4.2 Utilizing Sentiment Analysis

As mentioned in Section X.x, Sentiment Analysis is a useful domain for detecting the presence of issues. To test if this objective can indeed be satisfied, a number of tools were studied, alongside with the test documents above.

For example, Lexalytics' Sentiment Analysis tool [Reference] – Semantria - provides a comprehensive analysis of text, as shown in Figure 3.

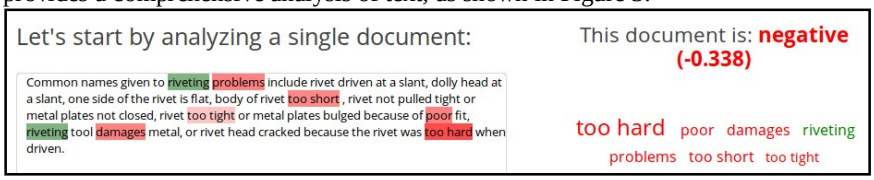

Fig. 3. Screen grab of Semantria's sentiment analysis

Similarly the performance of another openly available tool namely sentiment_classifier [Reference] was also tested to see whether it can be used as is for our purposes. However it was not trained separately and was used *as-is*, since the training exercise requires both domain resources and effort. Although many of the sentences were classified were classified correctly, the sentences which require domain background were not classified accurately. Shown is an example usage of this tool for "*Rivet head has some play*".

```
0. Rivet head has some                  ... pos = 0.25          neg = 0.0
-----------------------------------------------------------------------------
testfile.txt is Positive                ... pos = 0.25          neg = 0.0
Overall score of document
Total Pos = 0.25
Total Neg = 0.0
-----------------------------------------------------------------------------
```

Fig. 4. Example of using sentiment_classifier 0.6.

From sentiment_classifier, some example classifications were,

"*not properly lubricated*.", "*Bike squeaks when riding or pedaling*", "*rivet driven at a slant*" were all marked as neutral,

"*rivet head has some play*" was marked positive.

The conclusion from this exercise is that these sentiment analysis tools do a good job, with the exception of not being able to recognize domain-dependant tools. This could be improved by training such classifiers on a domain corpus, or by enhancing the lexicon used (in this case, SentiWordNet). The first method requires a large

amount of training data, and considerable manual effort. The second method is a current topic of research, which is discussed in literature [Yet to add].

Nonetheless, Sentiment Analysis is a promising means of identifying issue from text for us.

5 Discussion and Conclusion

This paper has introduced the objective of the research – to acquire diagnostic knowledge, and identified the need for a method to detect locations in text where such knowledge may be present. The detection of issues has been explained as the starting point. Some example documents have been studied and some observations regarding issues have been made.

Based on the observations, two possible methods are proposed – namely, one based on the functional descriptions of systems, and secondly, Sentiment analysis. The functional description based method has not been well explored. However, one apparent disadvantage with the method is the absence of large-scale resources such as function repositories and ontologies that serve a domain knowledge base. Hence until the time such knowledge becomes openly available, it is practically difficult to exploit.

The second method, Sentiment Analysis, is found to be more practical. A good number of tools and resources exist for using this set of methods. The method was tested with some domain examples, and performs reasonably well. However caution has to be exercised where the language is increasingly domain-specific. Since some part of the issue-based knowledge is dependent on context, a good amount of background knowledge in the form of domain related resources is necessary. This, in combination with sentiment lexicons, carried good potential for our work. Additionally, some specific needs have to be met. The method should be able to handle phrases also. Moreover, the identification of the issue is a starting point. Sentiment analysis will only tell us the polarity of the current text. The next step in understanding the issue is the identification of the cause(s) of the issue.

6 Future Work

The next steps would be to identify how to tune a lexicon for domain-dependant terms as well how to use such methods in conjunction with the logical form of text, if possible.

Acknowledgments. Acknowledgements are due to the authors and developers of the open tools that are being used in this work namely Boxer and C&C Tools, Kathuria Pulkit for Sentiment-Classifier.

References

1. Guidance for Performing Failure Mode and Effects Analysis with Performance Improvement Projects - QAPI
2. QUALITY BASICS: Root Cause Analysis For Beginners by James J. Rooney and Lee N. Vanden Heuvel, QUALITY PROGRESS JULY 2004 P.45
3. Contextual Valence Shifters, Livia Polanyi* and Annie Zaenenâ€
4. Function in Device Representation, B. Chandrasekaran and John R. Josephson
5. https://semantria.com/demo
6. Development of a Functional Basis for Design, Robert Stone and Kristin Wood
7. http://pythonhosted.org/sentiment_classifier/

POST_SCRIPTUM POINTS:

Identification of issues from logical form:

Function perspective requires large scale models - until such time ?

The purpose of the entire exercise to reduce the prohibitive effort involved in manually reading documents and encoding the acquired knowledge.

Phrase level sentiment is what we are looking for

Causality is possible in Sentiment Analysis ?

Goal of SA is to check for only SA - not tell what is the actual problem(s)... Nor is causality/implication covered.

Using Text Processing Techniques to Automatically Enrich a Domain Ontology, Paola Velardi, Paolo Fabriani Michele Missikoff