

Evaluation of Methods to Identify Assembly Issues in Text

N. Madhusudanan, B. Gurumoorthy, Amaresh Chakrabarti

▶ **To cite this version:**

N. Madhusudanan, B. Gurumoorthy, Amaresh Chakrabarti. Evaluation of Methods to Identify Assembly Issues in Text. 12th IFIP International Conference on Product Lifecycle Management (PLM), Oct 2015, Doha, Qatar. pp.495-504, 10.1007/978-3-319-33111-9_45 . hal-01377476

HAL Id: hal-01377476

https://inria.hal.science/hal-01377476

Submitted on 7 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Evaluation of methods to identify assembly 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 process plans. 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 presented. 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 for their suitability for detecting issues from documents. The first method is based on functional analysis of designs, which deems that an issue is the result of the violation of a function or a related parameter. The second is a Natural Language Processing technique called Sentiment Analysis that aggregates sentiments from individual words. The relative suitability of these 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 design and part manufacturing. However, a number of issues that affect assembly could be avoided at the design and planning stage itself, if appropriate knowledge for this is available *a priori*. Such knowledge of issues, from previous experience, may already be recorded in various forms in case studies, issue reports, change requests and other such documents. A more general manifestation of this challenge of making legacy knowledge available is the closing of loop between two product life-cycles. Knowledge and experience recorded during one product life-cycle must be made available to inform the next or any future product life-cycle. 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 many domains, and across various parts of the product life-cycle.

The final goal of the research reported in this paper aims to capture offline the knowledge present in these documents, and present an assembly planner with such

knowledge in a contextualised form. The general framework for this work is shown in Figure 1.

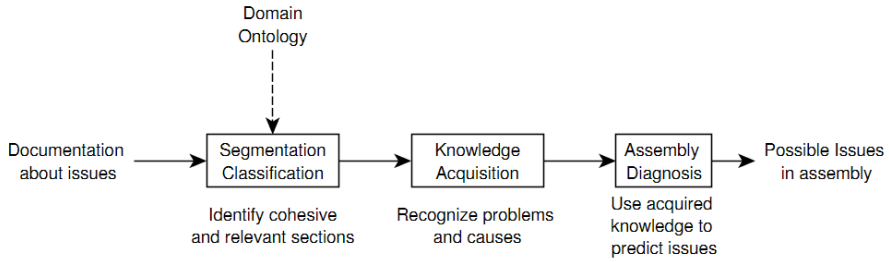


Fig. 1. A framework for acquiring diagnostic knowledge from documents

1.1 Issues in product design and realization

Issues can arise at any stage in a product's life-cycle. Examples of such stages include design [1], manufacturing, and deployment. The work reported in this paper is focused on the manufacturing stage, assembly in particular. Guidelines have been proposed successfully for tackling such problems (e.g. DFMA guidelines [2]). However, the applicability of generic guidelines is not without challenges [3]. The role of expert knowledge in design for manufacturing and assembly from various engineering areas has been stressed upon [3]. Such knowledge helps to cover uncommon, but, important issues, and is useful in domain-specific contexts. For example, these could be 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 manual-assembly intensive, and hence a large number of cases of many varieties are present, as opposed to assembly-lines, where few, specific classifications of problems are possible. Our aim is to make use of documents that contain instances of issues, and automatically acquire required diagnostic knowledge.

The paper is structured as follows. Section 2 looks at the objective of the paper and some existing methods for the same, followed by an initial study of documents to understand the representation of issues in text in Section 3. This is followed by exploration of two methods in Section 4 along with some examples. The conclusions of this exploration are presented in Section 5, followed by the future work in Section 6.

2 Current methods:

The objective of this paper is '*to explore and identify means for detecting an issue, wherever it is described in a text*'. In engineering diagnosis, methods for identifying faults and issues, or their causes, have been studied for some time now. The following review looks into methods that have been proposed for identifying issues and causes rather than representing these, such as in the case of Ishikawa / Fishbone analysis.

One such method, often applied in industry, is the Root Cause Analysis Method. It is a four-step process [4]: 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 Failure Modes and Effects Analysis (FMEA) [5]. 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 [6]. 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 [7] propose two types of diagnosis. In particular they point to the knowledge required for one type, namely malfunction hypotheses, and the relations between observations and malfunction hypotheses.

Farley [8] acquired cases for a diagnostic system from free text repair action message by building a grammar based on the possible patterns in the text. However, in this case, the messages were not exactly in natural English, but rather in a coded form. Also, the aim here was to acquire cases, rather than knowledge pertaining to issues.

Since we deal with text sources, natural language processing techniques were also studied. One of them is Sentiment Analysis, where the objective is to find the overall sentiment of a piece of text by analyzing the individual sentiments of its words. For example, Taboada et al. [9] calculate a semantic orientation based on sentiments of 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, in [9]). A general English network of words (SentiWordNet) that indicates sentiment has also been in vogue for some time now [10]. 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' Semantria are also quite effective [11].

Given the constraint of not having a very large data set, and the fact that our goal is to move beyond only 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-facto 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 a piece of text using this method. The functional representation and sentiment analysis methods are more promising from the perspective of this work. Both are explored in greater detail in the following sections.

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

3 Initial Study of Documents

The final goal of this work is to acquire diagnostic knowledge to a level at which it can be reused. To do so, the presence of issues and causes of issues in a text is important. The overall flow of the desired acquisition process is shown in detail in Figure 2. This schematic corresponds to the 'Knowledge Acquisition' step shown in Figure 1.

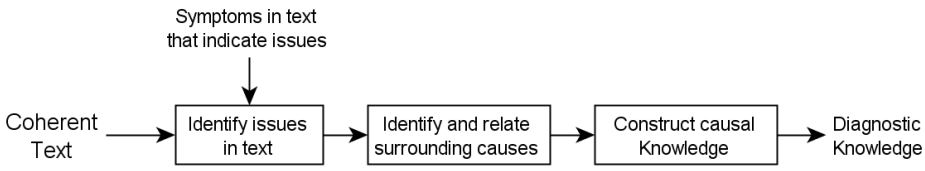


Fig. 2. Procedure to construct diagnostic knowledge.

In order to develop a means of acquiring diagnostic knowledge from documents, it was necessary to understand how such issues are recorded in text. For this, we surveyed a number of documents that are available in the public domain, and extracted portions that contain text where issues have been reported. A total of 20 such samples have been studied. These samples were from different domains, ranging from consumer complaints to bicycle maintenance, and riveting in aircraft assembly. Due to practical considerations such as corporate confidentiality, it is difficult to obtain documents from the industry. However, real examples, which are available in domains relevant to this work, have been carefully selected.

During the course of this study, the exact words (or phrases) which indicated the presence of an issue or an undesirable state were manually marked. Wherever it was found, 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 in the expression of issues and their causes in text emerges from such a study.

The following are some of the observations from the study:

- There are some 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 may be more than one issue expressed in a single sentence
  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 that one problem 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
- The amount of background knowledge required to understand the occurrence 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 - build up, or post-analysis or both. The causes of the issue may be explained by building up the final issue, the issue can be mentioned first followed by an explanation. And sometimes it can be a combination of the two.
- It is not yet known if the set of root causes is already known or unknown - the condition 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 the analysis of documents containing issues, our next step was to identify what methods could be employed to identify these automatically, and whether they could be used as is, or needed to be modified for our purposes. Before carrying this out, it is worth explaining a particular detail of our work.

The need for diagnostic knowledge dictates the need to acquire knowledge of issues as well as knowledge about causes of issues. This implies a need to understand the contents of the document. This need, as well as the subsequent use of a logical form to do so using Discourse Analysis method has been shown in an earlier paper [12]. Hence all such analyses of issues and causes in this work would be eventually performed on a logical form of text rather than the text itself. One example of a logical form (as given by the Discourse Representation Structure (DRS) tool Boxer [13]) is as follows. For the sentence

"*An aircraft has many parts.*"

x1, x2, x3 are the discourse entities, and the corresponding logical form is
patient(x1,x2), agent(x1,x3),have(x1), parts.(x2), quantity(x2), aircraft(x3);
(The patient and agent predicates are thematic roles for the entities (parts, aircraft) as expressed in the logical form. The 'patient' thematic role indicates something being affected, in this case, *having parts*. The other predicates indicate which variables are what, such as *x3* being *aircraft*.)

From the discussion regarding logical form presented above, we foresee two possible methods using which presence of issues in text could be identified in either text or its logical form. 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 Section 3, the presence of a domain related function or parameter can be a clue for 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 [14, 15,16]. These may be useful for detecting undesired behaviour of a system by means of negation of the function or any of its parameters. Literature exists about representing such functions [14], and building a basis of such functions for design [15]. Possible sources of function-related information include functional ontologies, such as Design Repository [17] and the SAPPhIRE-based ontology [18].

Compiling such ontologies and resources is labour-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 automatically in texts.

4.2 Utilizing Sentiment Analysis

As mentioned in Section 2, 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 the test documents above. For the purposes of the study, the input given to the tools were samples from a document that contained riveting issues. The document was one that is available in the World Wide Web. The output of such tools is an assessment of the overall sentiment polarity of the document / sentence / phrase (whichever is supplied). The polarity may be positive or negative.

For example, Lexalytics' Sentiment Analysis tool [11] – Semantria - provides a comprehensive analysis of text, see Figure 3.

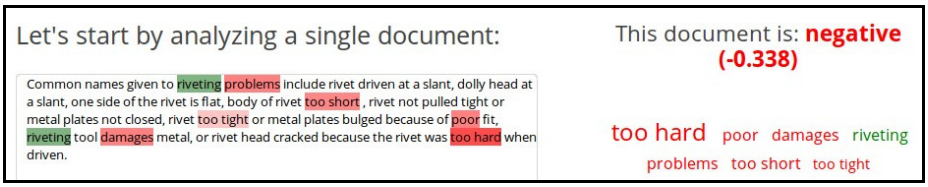


Fig. 3. Screen grab of Semantria's sentiment analysis

Similarly the performance of another openly available tool namely *sentiment_classifier* [19] was also tested to see whether it can be used for our purposes. However, it was not trained separately and was used *as-is*, since the training exercise requires both domain resources and effort.

```
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.

Using *sentiment_classifier*, we manually tested the performance for our examples. Here, complete sentences were not used as input, since we wanted to compare the manual identification of text that indicates issues, with the tool's output. Parts of sentences that were found to indicate the presence of an issue were used as input. These were phrases, or combination of two or more phrases. Some example phrases and their semantic classifications are shown below. The inputs were the phrases that are in quotes and italicised.

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

"*rivet head has some play*" (also shown in Figure 4)
"*riveting tool damages metal*" were marked *positive*.

"*metal plates bulged because of poor fit.",*
"rivet head cracked because rivet was too hard when driven."
"rivet could be too tight." were marked *negative*

The text *"rivet driven at a slant"* was also part of a larger text supplied to Semantria (Figure 3). In both cases, it was marked as neutral in sentiment. Similarly, for the text "*body of rivet too short.*", both Semantria and sentiment_classifier marked the text negatively.

Hence we see that some of the cases or parts thereof are covered correctly by such Sentiment Analysis tools. However, there are some domain-specific cases that do not give the desired output. For example, Semantria could not recognize the negativity in

"*one side of the rivet is flat*". Similar is the observation with sentiment_classifier shown above in the cases where they were marked neutral or positive (e.g. "*rivet head has some play*"). The possible reasons could be,

a) the absence of these words or phrases in the sentiment lexicon (or training data) with correct sentiment values ("rivet driven at a *slant*"), or

b) semantic ambiguity (e.g. for "*play*" – unless there is an explicit recognition of which meaning of "*play*" is in use, and if it has a negative sentiment assigned).

From the above exercise, we can observe that current sentiment analysis tools are practical and usable. However, the accuracy is not high while trying to recognize domain-dependant texts. We have only yet tested this on small, but representative pieces of text. Though this still requires testing on a larger scale, sentiment analysis has a greater practical appeal compared to functional ontologies. The performance could be improved by training such classifiers on a domain corpus, or by enhancing the lexicon used (in this case, SentiWordNet). The first approach requires a large amount of training data, and considerable manual effort. The second approach is a current topic of research in literature, and has been indicated as being harder to build than general lexicons [20]. For example, one such method [20] starts with an available sentiment lexicon and expands it using unsupervised methods. To do so, clauses that indicate positive or negative sentiment called *polar clauses* are looked for. Thereon, clauses of the same sentiment in successive clauses are acquired. Another method constructs aspect-specific domain lexicon by using a general lexicon and an opinionated corpus, and treating the final formulation as a linear programming problem [21].

Nonetheless, Sentiment Analysis, for us, is a promising means of identifying issues from text, because of the available tools and resources for doing so despite the need for adapting these resources in a domain-relevant manner. Also, it is also suitable for the kind of natural language text that we work with.

5 Discussion and Conclusions

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. 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 objective mentioned in Section 2 and observations drawn from documents about how issues are represented in text (See Section 3), two possible methods were explored – one based on functional descriptions of systems, and the other based on Sentiment analysis. The functional description based method has not been well explored. A disadvantage with this method seems to be the absence of large-scale resources such as function repositories and ontologies that can serve as 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 performed reasonably well. However caution has to be exercised in cases in which the language is highly domain-specific. Since some part of the issue-based knowledge is dependent on the context, a good amount of background knowledge in the form of domain related resources is necessary. This, in combination with sentiment lexicons, provided good background for our work. Additionally, some specific needs have to be met. The method should be able to also handle phrases. Moreover, identification of an issue is a starting point. Sentiment analysis will only tell us the polarity of the current text. The next step in understanding the issue would be identification of the cause(s) of the issue.

6 Future Work

We have discussed two possible methods to identify issues in a domain. This is one of several steps towards the eventual goal of constructing a diagnostic knowledge base.

At this point, Sentiment Analysis seems promising for identifying an undesirable state from a logical form. The next step is to implement and test the method. The implementation will require us to find a suitable means of extending the sentiment lexicon in a domain-specific manner, within our constraints. Some previous work about extending lexicons has been reported in Section 4.2.

To complete the reconstruction of an issue from text, it is necessary to realize the occurrence and causal relations using the logical form. Hence the current framework has to be extended for linking causes of the issue to the issue itself. It then has to be implemented and its performance tested.

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

References

1. Hales, C. "Five fatal designs." Proceedings of the International Conference of Engineering Design, ICED95. 1995.
2. Boothroyd, Geoffrey. "Product design for manufacture and assembly." Computer-Aided Design 26.7 (1994): 505-520.
3. Lahtinen, Tommi. "Design for Manufacturing and assembly rules and guidelines for Engineering." (2011).
4. QUALITY BASICS: Root Cause Analysis For Beginners by James J. Rooney and Lee N. Vanden Heuvel, QUALITY PROGRESS JULY 2004 P.45
5. Guidance for Performing Failure Mode and Effects Analysis with Performance Improvement Projects – QAPI, http://www.cms.gov/Medicare/Provider-Enrollment-and-Certification/QAPI/downloads/GuidanceForFMEA.pdf, accessed 28 May 2015.

6. “Fault Tree Handbook with Aerospace Applications’, Version 1.1, NASA Publication, August 2002.
7. Chandrasekaran, B., J. W. Smith, and Jon Sticklen. "‘Deep’models and their relation to diagnosis." Artificial Intelligence in Medicine 1.1 (1989): 29-40.
8. From free-text repair action messages to automated case generation, Benoit Farley, AAAI Technical Report SS-99-04.
9. Taboada, Maite, et al. "Lexicon-based methods for sentiment analysis." Computational linguistics 37.2 (2011): 267-307.
10. Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." LREC. Vol. 10. 2010.
11. https://semantria.com/demo, accessed 15 May 2015.
12. Madhusudanan, N., B. Gurumoorthy, and Amaresh Chakrabarti. "Segregating discourse segments from engineering documents for knowledge acquisition." Product Lifecycle Management for a Global Market. Springer Berlin Heidelberg, 2014. 417-426.
13. Curran, James R., Stephen Clark, and Johan Bos. "Linguistically motivated large-scale NLP with C&C and Boxer." Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics, 2007.
14. Stone, Robert B., and Kristin L. Wood. "Development of a functional basis for design." Journal of Mechanical design 122.4 (2000): 359-370.
15. Chandrasekaran, Balakrishnan, and John R. Josephson. "Function in device representation." Engineering with computers 16.3-4 (2000): 162-177.
16. Chakrabarti, A., and Bligh, T. P. A Scheme for Functional Reasoning in Mechanical Conceptual Design, Design Studies, 22, 6, 493-517, 2001.
17. Design Repository, http://function2.mime.oregonstate.edu:8080/view/index.jsp accessed 28 May 2015.
18. Srinivasan, V., Chakrabarti, A., and Lindemann, U. Towards an ontology of engineering design using SAPPhIRE Model, CIRP Design 2012: Sustainable Product Development, A. Chakrabarti (Ed.), Springer-Verlag, London, 2012.
19. Sentiment Classification using WSD, http://pythonhosted.org/sentiment_classifier/ accessed 28 May 2015.
20. Kanayama, Hiroshi, and Tetsuya Nasukawa. "Fully automatic lexicon expansion for domain-oriented sentiment analysis." Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2006.
21. Lu, Yue, et al. "Automatic construction of a context-aware sentiment lexicon: an optimization approach." Proceedings of the 20th international conference on World wide web. ACM, 2011.