Mining Automotive Warranty Claims Data for Effective Root Cause Analysis

Ashish Sureka, Sudripto De, and Kishore Varma

Infosys Technologies Limited,
Bangalore 560100, India
{Ashish_Sureka,Sudripto_De,Kishore_Varma}@infosys.com

Abstract. We present an application of text analytics in automotive industry and describe a research prototype for extracting named-entities in textual data recorded in automotive warranty claim forms. We describe an application for gaining useful insights about products defect reported to the dealer during the warranty period of vehicles. The prototype is developed for air-conditioning subsystem and consists of two main components: a text tagging and annotation engine a query engine. We present some real world examples with sample output and share our design and implementation experiences.

Keywords: Text analytics, product warranty data analysis, text tagging and annotation.

1 Introduction

Product defects and warranty claims results in heavy costs to manufacturers. The top 50 U.S.-based warranty providers together reported $23.0 billion in warranty claims during 2006, up 5.1% from 2005. It is interesting to note that auto manufacturing companies in USA such as General Motors Corporation and Ford Motors Corporation are amongst the top 50 U.S.-based warranty providers of product warranties in terms of the total dollar amounts they reported in warranty claims during calendar 2006. GM and Ford together spent $8.6 billion on claims during 2006 and auto manufacturers grab the bulk of the pie in the overall extended warranty market. Auto manufacturing companies are spending a huge percentage (around 2.5%-3.0%) of their sales revenue fixing vehicles under warranty which puts a tremendous pressure on auto manufacturing companies to come up with innovative ways to reduce the overall cost to the company on warranty claims by reducing the detection to correction time of a product failure and also to increase customer satisfaction and brand value [7][9][11].

More importantly, the famous Bridgestone and Firestone's recall of 6.5 million tires used primarily on Ford Explorer vehicles, and the deaths of more than 100 people in accidents blamed on the failure of those tires triggered legislators to introduce a new U.S. law called as Transportation Recall, Enhancement, Accountability and Documentation (TREAD) Act which makes it mandatory for automakers to compile quarterly reports on consumer complaints and warranty claims. Recall of

J.R. Haritsa, R. Kotagiri, and V. Pudi (Eds.): DASFAA 2008, LNCS 4947, pp. 621–626, 2008.

vehicles due to safety related product defects costs a huge amount of money to auto makers and has become a very serious issue because of the loss of human life and injuries as a result of those defects. Another famous example is recalls of around 800,000 Jeep Liberty vehicles because of a defective steering part. The defect caused drivers to lose control of their vehicle and suffer serious injuries in road accidents. The US Consumer Product Safety Commission provides public access to data on recalls under various product types, companies and date [4][6][8][10][12].

The pressure to reduce spending on warranty claims and compliance to legal regulations has prompted automakers to look more carefully into warranty claims data. The business driver for the work presented in this paper is to build tools and techniques that can help discover defects early in the product life-cycle and enable a warranty analyst to identify root causes of failures by leveraging textual data in conjunction with the structured data recorded in claim forms.

1.1 Textual Data in Product Warranty Claim Forms

Automotive warranty data is generally gathered by filling a claim form by a customer and a technician. The form is either filled on a paper which is later scanned and imported into a database or the information is directly entered online. A form can contain many fields to be entered by a customer or a technician. Some of the fields require information such as the product code, model number, date and time-stamp and customer id. This information falls into the category of structured data in the sense that the information has a well defined format and requires close-ended answers i.e. there are finite choices from which a selection can be made. Usually the form also contains a comments section where a customer or a technician can provide detailed information about the problem. The comment section is provided as it is not possible to capture details about a defect using the structured data fields alone. This is the section where information is entered in the form of a natural language text or a free-form text. The data entered in comment sections is a key element in diagnosing and understanding the problem. Following are some of the examples of the customer complaint, technician comments and action taken field data.

Customer complaint
- Air conditioning not working
- Poor performance from a/c system
- Water ingress into passenger foot well
- Room lamp flickering when switched ON

Technician Comments
- Found expansion valve defective
- Thermostat by-pass valve not working
- No engine cranking noise. Solenoid check.
- AC Knob found broken

Action Taken
- Removed and replaced the seals on the elbow joints
- Cable set properly & refitted

- AC cable replaced resulting smooth movement
- Connected the coupler to compressor

If the auto manufacturer suspects a recurring problem they sift through the claims data and manually go through the customer and technician comments to see if they can find any kind of patterns or clue that can help them finding the cause of the problem. A high level analysis of a defect can be done from data stored in structured data fields. However, a drill down analysis or an in depth analysis requires a warranty claim analyst to read the free-form textual data fields also. The main challenge is that the manual process of reading each and every comment is impractical and time consuming. Hence there is a strong need in automating the process of analyzing the natural language text data stored in claim forms for an efficient data analytics**.**

2 Solution Approach

The end user of the system that we developed is an automotive warranty analyst who is primarily a domain expert belonging to the quality department of the automaker. The system has been designed keeping in mind the requirements outlined by a warranty analyst. One of the primary requirements of the warranty analyst was to have a graphical user interface based system where he can query the unstructured data (customer complaints and technician diagnosis expressed in free-form textual format) using high-level or natural language queries and generate reports. Hence, the system was designed to have the following two main components.

1. Text Tagging and Annotation Engine
2. Query Engine

We divided the process of mining warranty claims data into two phases. Phase 1 consists of converting free-form text data in warranty claim forms into structured data using a natural language processing technique called as named-entity extraction. Phase 2 is of reporting and analytics where a warranty analyst queries and analyzes the structured data obtained from Phase 1 process using high level queries. Figure 1 presents the high level architectural diagram illustrating the data flow and the two phases.

Text Tagging and Annotation also called as Named Entity extraction forms an important component of many language processing tasks such as text mining, information extraction and information retrieval. Named Entity extraction consists of identifying the names of entities in free-form or unstructured text. Some of the common types of entities are proper nouns such as person names, products, organization, location, email addresses, vehicle, computer parts and currency, temporal entities such as dates, time, day, year, month and week, numerical entities such as measurements, percentages and monetary values. There can be numerous domain specific entities also [1][2][3][5].

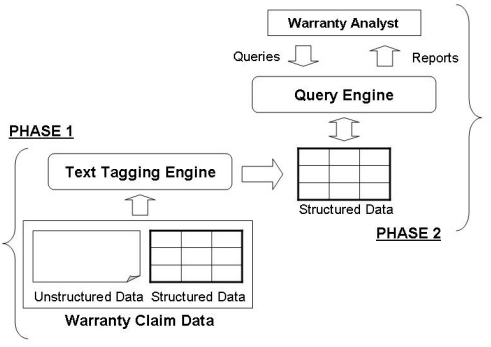

Fig. 1. High-level architectural diagram illustrating the two stages of the analysis

We developed a rule-based system for extracting named entities from customer complaint, technician comments and action taken field of the warranty claim forms. Some of the named-entities that we identified are technician action, car part location of a defect, reason of failure, effect of failure, defect type, condition under which defect occurred and customer action that caused the defect. Table 1 and 2 give examples of some customer complaints and the named entities extracted by our tool. Table 1 and 2 are for illustration purposes as it is not possible to present all the named entities with examples due to limited space in the paper. We made use of lookup tables to increase the accuracy of our system. The tagging and annotation engine is based on hand-crafted rules and lookup tables containing domain terms and clue words or phrases. Following is a simple example of a rule to illustrate the technique. Action taken by a technician is an entity that we wanted to extract from the action

Table 1. Customer complaints in warranty claim forms

Customer Complaints	
ID	**Complaint**
WCF01	Vehicle does not start when cranked
WCF02	Vehicle causes noise from below during turning
WCF03	Horn is not working
WCF04	When vehicle is stopped, lot of noise from engine
WCF05	Room lamp flickering when switched ON

Table 2. Output of tagging and annotation

Tagged Data for Customer Complaints			
ID	**Component**	**Problem**	**Customer Action**
WCF01	-	does not start	when cranked
WCF02	-	causes noise from below	during turning
WCF03	Horn	not working	-
WCF04	Engine	lot of noise	When vehicle is stopped
WCF05	Room Lamp	flickering	when switched ON

taken field in warranty claim forms. Some of the examples of technician action are replaced (replaced fuse or horn relay replaced), removed (removed evaporator unit), refitted (refitted AC able) and cleaned (cleaned the duct of AC system) etc.

It is practically not possible and scalable to create a lookup table of all the actions that a technician can take and hence we implemented a rule which scans each word in a sentence and checks if it ends with "ed" or "ing" If the word ends with "ed" and "ing" in technician action taken field then there are good chances that it is an entity of type action. However, just this rule is not enough as the word connected ends with "ed" but may not be an action in the context of the sentence "replaces belt connected to pulley". Hence some more levels of check or a chain of rules is required to identify an entity of correct type and disambiguate it from other entities. For instance, in the example "replaces belt connected to pulley", connected is not a technician action as it is also preceded by a word "to". The complexity of the rule depends on the type of entity that needs to be extracted and also depends on the writing style.

We did an evaluation of two popular text mining toolkits, GATE (General Architecture for Text Engineering) and LingPipe to see their fitment with the problem at hand [3]. To perform named-entity detection, LingPipe requires a supervised training of a statistical model and once a model is built it can be used to detect named-entities on unseen data of the similar nature as the training data. The training data must be labeled with all of the entities of interest and their types. We had around 250 sample data points which was not enough for us to select a machine learning based approach. The dataset available to us was limited, but all the 250 data points we had were of different types without any duplicates. Machine learning based approach is successful when there is a good quantity of annotated corpora to train a model. We tried using LingPipe but we were not getting good results due to insufficient training data. We will again evaluate LingPipe in the future when more sample data becomes available to us. GATE was another alternate and provides a mechanism to write hand-crafted rules and regular expressions in the form of pattern specification language called as JAPE (Java Annotations Pattern Engine) grammar. We tried writing rules using JAPE but realized that the heuristics required for extracting entities are easier to code in a programming language like Java as it required operations like finding the presence of a substring in a string, usage of features from previous annotations, usage of *if-then-else* statements and nested *for* loops. GATE provides functionality to call Java code from JAPE rules but we realized that the majority of our rules require flexibility of a programming language like Java and we found it easier to write our own custom code rather than calling our Java code from within GATE environment. Moreover, textual data in claim forms contain language which has lots of spelling mistakes, grammatical errors and short forms and that is why we chose to implement a custom rule-based system. There are two kinds of approaches to named entity recognition. One approach is based on using statistical modeling or machine learning whereas the other approach is based on developing rules and heuristics. We evaluated both the approaches in the form of LingPipe and GATE. However, our requirements were such that we finally decided to implement a custom rule-based named entity recognition system. We also created our own gazetteer list and pattern matching rules. Our gazetteer consists of two types of text tokens. One type of gazetteer list consisted commonly occurring terms such as vehicle parts (condenser, cooling coil etc) and technician actions (removed, replaced, adjusted). The other type of gazetteer list

consisted of trigger words. Trigger words are text tokens that provide indication or clue for an entity occurrence such as the presence of token “on” for a location entity.

In the prevailing circumstances where million claims being filed per annum, it becomes practically impossible for any warranty analyst to go through the text of the claims manually. As a result most of the info reported in the text goes unnoticed and undecipherable. Text analytics will help integrate and automate the process of deciphering information from text resulting in more effective defect discovery.

References

1. Tan, A.-H.: Text Mining: The state of the art and the challenges. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574, pp. 65–70. Springer, Heidelberg (1999)
2. McCallum, A.: Information Extraction: Distilling Structured Data from Unstructured Text. Social Computing 3(9), 48–57 (2005)
3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia (July 2002)
4. Batesa, H., Holwegb, M., Lewisc, M., Oliverd, N.: Motor vehicle recalls: Trends, patterns and emerging issues. OMEGA: International Journal of Management Science 35(2), 202–210 (2007)
5. Zhang, L., Pan, Y., Zhang, T.: Focused named entity recognition using machine learning. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 281–288 (2004)
6. Fournier, R., Shovelton, T., Stolle, L.: Walking the automotive industry tightrope: Keeping customer and your brand safe, An Executive strategy report of IBM Global Services published on (April 14, 2003)
7. Teret, S.P., Vernick, J., Mair, J.S., Sapsin, J.W.: Role of Litigation in Preventing Product-Related Injuries. Epidemiological Reviews 25, 90–98 (2003)
8. Automotive Warranty Management: Paying the Bill and Solving the Problem by Kevin Prouty, A report on Manufacturing, AMR Research(November 01, 2000)
9. Recalls and Product Safety News, U.S. Consumer Product Safety Commission, http://www.cpsc.gov/cpscpub/prerel/prerel.html
10. The Warranty Process Flow within the Automotive Industry: An Investigation of Automotive Warranty Processes and Issues. Center for Automotive Research (August 2005)
11. Warranty Week, The Newsletter for Warranty Management Professionals, http://www.warrantyweek.com/