Unsupervised Extraction of Part Names from Service Logs

Aravind Chandramouli, Gopi Subramanian, Debasis Bal

Abstract— **Service logs are created by service engineers while resolving issues with industrial equipment. These service logs contain valuable information that is used by downstream applications but are highly unstructured. One such information in the service logs is the part names that are hard to identify because they are often misspelt and are also used differently depending on the service engineer creating the service log. Supervised techniques are used in industrial domains for extracting business intelligence. These techniques have the drawback that it is expensive to create training data and hence, we explore the use of unsupervised technique based on unigram and bigram language model to extract part names from these service logs. We evaluate our approach on over 4,500 part names and our approach achieves a Precision of 76.9%.**

Index Terms- **language model, unsupervised technique, business intelligence, text mining**

I. Motivation

In industrial systems, the underlying databases are well structured to facilitate capture of raw text under various categories. For example, in a service center, a database might consist of the following categories: the machine name, parts list, issues with the parts and the fix made for the issue. However these names and lists tend to go obsolete with newer installation and engineering design changes. Industrial equipment like turbines, generators and others are made up of thousands of parts. During field operations, it is highly impossible for a service engineer to correctly identify the normalized name for a part under inspection or to find the serial number of the part, while mentioning a problem about the part. Similarly, a fix might be a made by another engineer who might use his/her own terminology in referring to the part name and the solution that was used. The part names, customer complaints, issues identified and the fix made are usually documented in free text in an unstructured format called as service logs. Hence, a critical activity for downstream applications is the extraction of information from these free text fields to specific categories of interest.

The specific problem that we address in this paper is the identification of part names from the service logs. The primary challenge is that in addition to being unstructured, the service logs contain a lot of non-words, domain specific abbreviations and jargons and spelling errors (In many cases, the misspelt word is the correct way of representing that word). For example, stage 10 compressor is referred in the following manner: stg 10 compressor, stage ten compressor, stge 10 compressor, stg10 compressor. These part names are used downstream for applications like building early warning models for failures and similar business intelligence applications. There are a number of approaches that can be used to extract the part names. Supervised learning techniques have been used with great success for similar tasks [Yu et al. 2005]. One major disadvantage is the necessity to create a training set with annotations that identify the part names which is a time consuming process and even domain experts will have a tough time identifying some part names since the terminology used is diverse. For the reasons discussed above, manually creating a knowledge base of part names, with all their variants is not possible. Hence, we explore the suitability of using unigram and bigram language models to identify these part names. In Section II, we look at related work and we describe our approach in detail in Section III. Section IV describes our results and we finish with a few concluding remarks in Section V.

II. Related work

Named Entity Recognition (NER) refers to systems that identify entities of interest and these entities of interest vary depending on the domain. Like any other information extraction task, the systems can be broadly classified as either unsupervised or supervised. Since our method is unsupervised, we restrict our literature survey to unsupervised NER systems. Unsupervised techniques tend to depend on an external lexicon, such as WordNet or lexical patterns. For example, Alfonseca and Manandhar [2002] attach topic signatures to known concepts from WordNet. For an unknown concept, a topic signature is generated using the neighboring words and this is compared with the topic signatures of the known concepts using a similarity function to predict a concept. Instead of the WordNet, Turney [2001] uses the World Wide Web for finding the appropriate synonym for a given word by issuing the word and the list of candidate synonyms as query to the AltaVista search engine. Co-occurrence between the word and the candidates are calculated using different variations of the PMI-IR algorithm. These variations are based on the definition of co-occurrence, for example, the words occur

The authors are with GE Global Research, John F Welch Technology Centre, EPIP Zone, Bangalore, India, 560 066.
E-mail: {aravind.chandramouli, gopi.subramanian, debasis.bal}@ge.com

close to each other in the same document versus the words occur in the same document irrespective of their location. KNOWITALL system [Etzioni and Oren, 2005] also uses the World Wide Web to extract named entities and PMI-IR is used as a feature to classify the named entities to a given category.

Another aspect of our work is the extraction of business intelligence from textual data and a number of research efforts have been focused on this. Yu et al. (2005) utilize text mining to construct a metadata repository from unstructured text and rough set theory is then used on the constructed knowledge base for forecasting crude oil market tendency. To demonstrate the effectiveness of their system, they compare their approach with other forecasting models based on linear regression, neural network etc. Kornfein and Goldfarb (2007) look at the issue of categorizing manufacturing quality defects using a data set that is similar to ours in terms of content and quality. They compare a machine learning approach with a rule based algorithm for classification and found for their data set that the accuracy of the rule based algorithm was very close to that of the best performing learning algorithm. Another application of text mining for categorization is by Huang and Murphey (2006), who look at assigning problem descriptions to diagnostic categories for automobiles. In the civil engineering domain, Caldas et al. (2002) look at automatically classifying construction project documents and found that SVM was the best performing classifier in their experiments. They further extend their work in Caldas et al. (2003) and look at using hierarchical classification for managing documents in the construction management information systems.

III. Approach

Figure 1 shows the architecture of our system.

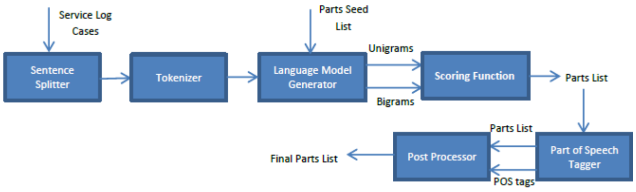


Figure 1. System Architecture

The service logs serves as the input to the SentenceSplitter and the generated sentences are passed onto the Tokenizer module that splits the sentences into tokens. In addition to the tokenization, the Tokenizer module also removes stop words like the, an, a and the tokens are then passed to the LanguageModelGenerator module that takes an additional input, the Parts seed list. This seed list contains the parts of interest like valve, bearing etc. The LanguageModelGenerator works by using the hypothesis that part names act as the prefix to the parts of interest, for example, if valve is a part of interest, the part names associated with parts of interest could be control valve, bypass control valve, bypass ctrl valve, bypass ctl valve. Control, ctrl and ctl all act as synonyms but they are used as prefix to the part of interest. For “Control valve”, control is an example of a unigram generated and “bleed control valve”, bleed control would be an example of a bigram that is generated by the module.

The generated unigrams and bigrams serve as input to the ScoringFunction module. The ScoringFunction produces a ranked list of the n-grams as possible part names. A score is assigned to each input n-gram by calculating the correlation between the n-gram and the suffix. This correlation is estimated using a modified version of Mutual Information and the formula is given below:

$$\mathrm{I(G, S)} = \log \frac{\frac{C(G,S)}{N}}{\frac{C(G)}{N} * \frac{C(S)}{N}} * smoothingCoefficient$$

$$\text{where } smoothingCoefficient = 2 * C(G,S)$$

N = Number of words in the corpus.

In the formula above, C(G) refers to the count of the n-gram G in the corpus, C(S) refers to the count of the suffix S and C(G, S) refers to the co-occurrence count of G and S. Here, *S* refers to the suffix which is the seed part name and *G* refers to the n-gram that is generated by the LanguageModelGenerator.

The n-grams are sorted by the scores computed above and the ranked list of part names is produced. The ranked list of part names thus produced is then tagged with their part of speech tags using a Part of Speech tagger. The PostProcessor module takes the parts list and their associated part of speech tags and produces a final ranked list of part names that contain the tag *‘NN’*.

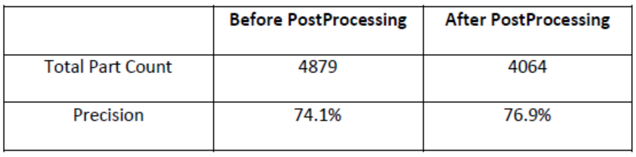
IV. Result

The algorithm described above was implemented and we used the list of seed part names that business had provided for our evaluation. There were a total of 433 seed part names and after removing duplicates there were a total of 130 seed part names for which there were mentions in the service logs.

To understand the quality of our algorithm, we look at Precision of the extracted part names. In this context, Precision is defined as follows:

$$\text{Precision} = \frac{Number\ of\ relevant\ Part\ Names}{Total\ number\ of\ Part\ Names\ retrieved}$$

Table 1. Algorithm Precision



	Before PostProcessing	After PostProcessing
Total Part Count	4879	4064
Precision	74.1%	76.9%

We looked at the Precision for the part list before and after the post processing step. The overall Precision for the system was 76.9%. The post processing step using part of speech tagging helped improve the Precision from 74.1% to 76.9% (an improvement of 2.8%). We also found the choice of the tokenizer had an impact on overall Precision due to the quality of tokens that were produced.

V. Conclusion

An efficient method to reliably extracting these part names from unstructured service logs has been described in this paper. The results (Precision of 76.9%) show the effectiveness of this approach and this approach can be extended to extract other interesting entities from service logs. We plan to use the extracted information for building an early warning model for part failures and also to extract other information like common issues associated with the parts, typical resolutions to build applications that will help service engineers quickly resolve issues.

References

[1] L. Yu, S. Wang, and K.K. Lai, A rough-set-refined text mining approach for crude oil market tendency forecasting International Journal of System Sciences 2(1) (2005) 33-46.

[2] M.M. Kornfein and H. Goldfrab. A comparison of classification techniques for technical text passages. in World Congress on Engineering. (2007) London, UK: Proceedings of the World Congress on Engineering.

[3] L. Huang and Y.L. Murphey. Text mining with application to engineering diagnostics. in 19th International conference on Industrial Engineering and Other Applications of Applied Intelligence, IEA/ AIE 2006. (2006): LNAI 1309-1317.

[4] C.H. Caldas, S.M. Asce, L. Soibelman, M. Asce, and J. Han, Automated classification of construction project documents, Journal of Computing in Civil Engineering 16(4) (2002).

[5] C.H. Caldas, L. Soibelman, Automating hierarchical document classification for construction management information systems, Automation in Construction 12 (2003).

[6] Alfonseca, Enrique, and Suresh Manandhar. "An unsupervised method for general named entity recognition and automated concept discovery." Proceedings of the 1st International Conference on General WordNet, Mysore, India. 2002.

[7] Etzioni, Oren, et al. "Unsupervised named-entity extraction from the web: An experimental study." Artificial Intelligence 165.1 (2005): 91-134.

[8] Turney, Peter. "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL." (2001).