Information Extraction for Lightning Strike Related Aircraft Maintenance

Ankita Mathur
Boeing Research & Technology
The Boeing Company
Bengaluru, India
ankita.mathur@boeing.com

Halasya Siva Subramania
Boeing Research & Technology
The Boeing Company
Bengaluru, India
halasyasiva.subramania@boeing.com

Micah Goldade
Boeing Research & Technology
The Boeing Company
Seattle, WA, USA
micah.l.goldade@boeing.com

***Abstract*— Lightning strikes can cause extensive damage to an aircraft and affect both maintenance and safety operations. Understanding the effectiveness of current lightning protection and zoning in aircraft is essential; subject matter experts (SME) use this understanding to develop actionable threat mitigation strategies for improving design and developing efficient post-lightning strike repair specifications. We have proposed the use of data analytics in order to create a consolidated data source for lightning strike related events from maintenance logs. We have used a dictionary-based named entity recognition and dependency-graph based relationship extraction in order to extract most desired information from maintenance logs.**

Keywords—information extraction, named entity extraction, relationship extraction, lightning strike, dependency graph model

I. Introduction

Lightning strikes can cause significant damage to aircrafts under certain circumstances and can cause costly delays and disruptions to airlines. The extent to damage caused by lightning strikes can range from no damage to severe damage. [1] Understanding the effectiveness of current lightning protection and zoning in aircraft is essential; subject matter experts (SME) use this understanding to develop actionable threat mitigation strategies for improving design and developing efficient post-lightning repair specifications. Additionally, knowing which parts are prone to lightning damage, and which airports have more lightning strike reports, will enable service industry to source spares and repair kits strategically.

Aircraft maintenance records are written by maintenance technicians describing their observations and actions. The level of detail in the maintenance logs are very subjective and dependent on the technicians. The reports are written in free-form text and SMEs put in a lot of manual effort to read through the text in its entirety, collect relevant information and understand the type of damage and the maintenance actions that were taken to fix. Natural language processing (NLP) and text analytics can extract relevant information from unstructured text and process it to consolidate information as per SME needs. These methods can help automate information extraction for a fleet of any aircraft model in few minutes, in turn improving SME productivity. Further, advanced analytics on fleet data can provide insights into implications from existing design.

This paper describes a technique for information extraction from maintenance data using language dependency graphs on a dictionary-based entity extraction algorithm. Using this technique we extract damaged part names, part locations, damage conditions and repair actions carried out by maintenance after the event of a lightning strike on an aircraft.

In the realm of Information Extraction (IE), named-entity recognition and relationship extraction are two significant sub-problems [2]. A named entity is a proper noun of significance. In principle, a named entity recognition (NER) task consists of identifying a proper noun and then classifying that noun into an entity type of significance [3]. Named entities can be of various types (class) and can be defined per specific task.

One method of named entity recognition (NER) is a dictionary-based approach which involves the use of a “lexicon” containing a vocabulary of words that might appear in the named entity. This approach is popular for NER in areas like medical science and genetic engineering where named-entities are domain-specific. Aerospace industry is no different, the part names and other terminologies are very domain specific and hence a dictionary based approach was the only choice, given there was no domain ontologies or lexicons available. However, there are a couple of drawbacks to this approach [4]: (a) High number of false positives due to short names (b) Spelling variations due to misspelling or regional spelling differences pose a huge challenge.

On the other hand, relationship extraction is another area of active research in information extraction, specifically used to establish meaningful semantic correlations between two or more words or entities. Detecting these meaningful relationships help in deriving insights based on situational context of the semantic usage. Two approaches are widely used for relationship extraction – (i) ontology; and (ii) visual parsing of entities based on parametric values associated to context and content. This paper uses the approach of using dependency parse trees which has been employed extensively in biomedical fields [2] [5].

Named entities can be of various types (class) and can be defined per specific task. For this paper we have defined the following classes of entities: (a) Part Name (b) Location on aircraft (c) Damage conditions (d) Lightning strike indicators and (e) Repair actions. We have used a dictionary-based approach for identifying these entities as the list of part names and locations are available to us, and the repair actions terminology is also finite.

Further, we explore the relationship between entities, so that to extract complete information of the lightning strike related incident. We have used a dependency graph based approach for finding relationships between entities and hence extract entity triples of (location, part, action) or (location, part, indicator) based on the available information. A consolidated data source created from these triples can be used to search for answers to many SME questions like, the location where most lightning strike occur, what type of actions are generally carried out to repair lightning related damage, or the level of damage occurred at various locations during a lightning strike event etc.

II. Data Driven Method

A. Overview

The design of text analytics pipelines is ubiquitous and consists of sequential text analytics tasks (tokenization, spell correction etc.) where output of one task becomes input to the next task [6]. The steps involved in our information extraction pipeline are shown in Fig. 1.

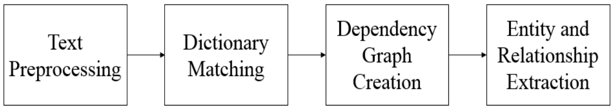

Figure 1 Information Extraction Pipeline

The techniques involved in every pipeline are different depending on the characteristics of the data. For example, sample maintenance log extracts are shown in Fig. 2. True to being unstructured, maintenance log data has text containing numerous abbreviations, misspellings, short names and grammatical errors.

Complaint Text	***Maintenance Text***
ENG LH REV TRAILING EDGE LIGHTNING STRIKE DAMAGE AT 7 O/C POSN	REPAIR C/O AS PER REPAIR MANUAL. SUBJECT DAMAGE
POSSIBLE LIGHTNING STRIKE NEAR RIGHT-HAND SIDE FUSELAGE	ACTION: FND LIGHTNING STRIKE AT WBL 1184 STA JUST BESIDE THE STATIC DISCHARGER AT THE NAV LT.

Figure 2 Example maintenance logs

Therefore, data preparation and preprocessing of the text becomes the first task. In this task, the text is cleaned for misspellings, abbreviations are expanded and the text tokenized as words, pairs or n-grams. Subsequently, the words (tokens) are matched to the dictionaries of parts, locations, actions and lightning strike indicators' dictionaries. This helps in identifying candidate entities segmenting the text respectively. In the next task, after the candidates are tagged with entity class name, we create a dependency graph for each sentence in the text which shows the relationships between words (tokens). An example of such a graph is shown in Fig. 3.

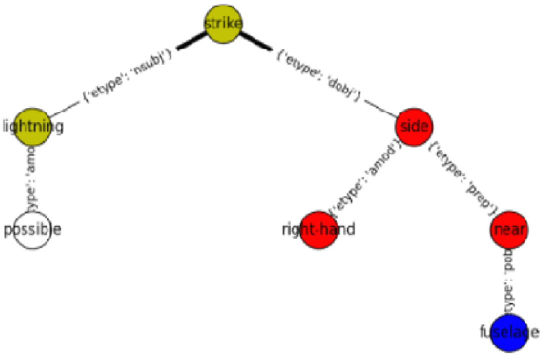

Figure 3 Example of a dependency graph

In the final task, we extract triples of (location, part, action) or (location, part, indicator) using the patterns of dependency graphs frequently seen with respect to relationships between the different entities.

B. Text Preprocessing

There are three sub-tasks involved in this task: (a) Tokenization, (b) Spell corrections and (c) Abbreviation expansion.

We use regular expressions to tokenize the data instead of using standard word tokenizers. This was needed in order to capture more detailed domain-specific information. Spell correction of complaint and maintenance text is performed using a "context sensitive" method [7]. Isolated-term correction would fail to correct typographical errors such as "lightening strike on", where all three terms are correctly spelled English words but are wrong in usage of the words surrounding each other given the context.

The algorithm follows a simple approach by using n-grams for spell correction. A probability distribution of bigrams is created beforehand using all text data available and stored in a file. When the algorithm determines that a word must be corrected, a trigram is passed to the spell correction module along with a list of possible corrections. The context sensitive algorithm substitutes the word in scrutiny with words from possible correction list and compares the probability of the new phrases to find the most suitable correction. Using the bigram probability distribution we compute the probability of the trigram by using equation (1). The trigram with the highest probability is chosen to correct the word.

$$P(w1, w2, w3) = P(w2|w1) * P(w3|w2) \qquad (1)$$

Abbreviation expansion is performed by the algorithm to include more detail in the text. A list of abbreviations and their expansion is created manually by observing a volume of data. For example: FWD is an abbreviation used for the word FORWARD. The abbreviations are substituted by their expansion.

C. Dictionary Matching

The lexicon of part names and locations on an aircraft were generated from proprietary data sources. For actions and indicators, lexicons were created using commonly used words in the maintenance logs data to describe these situations. Some examples of action words are *repair, replace, inspect etc.* Some examples of indicators are *lightning strike, burn marks, lightning encounter etc.*

We used these lexicons to mark potential words as named entities. As shown in Fig. 3, *lightning strike* is an indicator, *right-hand side near* is a location and *fuselage* is a part name. We used a fuzzy string matching algorithm to match the words to the dictionary. The fuzzy string matching function uses Levenshtein Distance between a candidate word to be matched and all the dictionary words to calculate similarity and label the word as "part".

For example, consider the following token set:
["lightning", "strike", "on", "fuselag", "sta890"]

With normal string matching "fuselag" will not be identified as a part. But with fuzzy string matching it will be identified as part because similarity between "fuselag" and "fuselage" is high. The similarity between candidate word and dictionary word should be above a threshold in order for them to be considered a match. The threshold criterion is set based on an empirical study on a sample dataset.

D. Dependency Graph Creation and Relationship Extraction

The technique used for relationship extraction in this paper is graph processing of language dependency graph [8]. The graph lets us analyze the structure of the sentence and extract information easily. We use Stanford dependency parser [9] to create dependency graphs. The graph quickly lets us use the Stanford dependencies to determine which parts we affected by lightning and what repair actions were carried out. Within the graph of the sentence we look for a relationship between part-name word and location word to link part-name and location. In Fig. 3, "RIGHT-HAND SIDE NEAR" and "FUSELAGE" are related by relationships "amod (adjective modifier)","prep (preposition)" and "pobj (object of a preposition)". Moreover, the location is related to strike indicator "Lightning Strike" by the relationship "dobj (direct object)". Such relationships are frequently seen in the dataset for lighting strike related events. Our method exploits these frequently seen relationships in the form of rules in order to extract the desired triples as shown in Fig. 3.

The final output from Fig. 3 is the triple ("RIGHT", "FUSELAGE, "LIGHTNING STRIKE")

III. Results

A sample of 1000 maintenance logs were taken from the dataset in order to conduct the experimental study. The sample was then tagged for named entity and relationship triples manually by SMEs. The dataset was then processed using the pipeline described above and the results were compared with the manual results. For the experiment we used 4 different lexicons with different sizes to extract part-names. Lexicon A had 4719 words, Lexicon B had 4800 words, Lexicon C had 5220 words and Lexicon D 9995 had words. The precision and recall [10] values for each lexicon was as shown below:

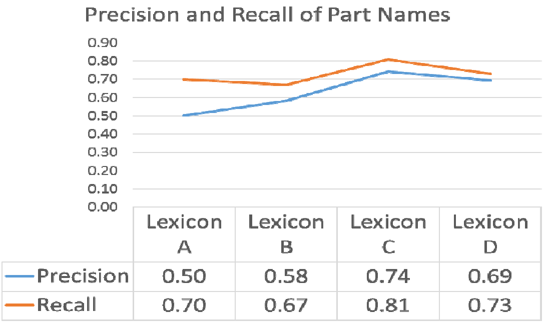

Figure 5 Precision and Recall of Part Names

The precision and recall for all classes of named entities are as shown below: The values are consistent with requirements of the SMEs that want high precision classification.

TABLE I. Precision and Recall of Various Entity Types

Entity Type	Precision	Recall
Part	0.74	0.69
Location	0.91	0.65
Action	0.92	0.74
Indicator	0.99	0.96

Another experiment was conducted to check the significance of the context-based spell correction and the results for part-name class are as shown in Fig 5. The results show that use of context based spell correction is effective to increase accuracy of the algorithm in finding various classes.

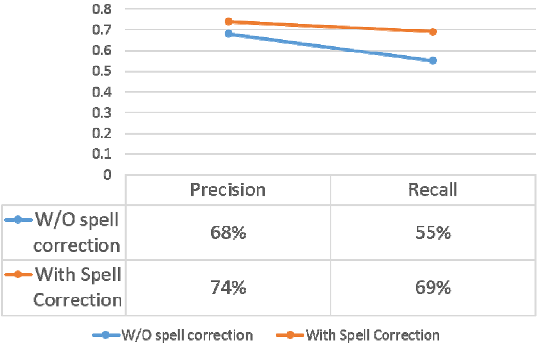

Figure 4 Precision and recall with respect to spell correction

IV. Conclusion

The information extraction technique used in this paper effectively disentangles an event of lightning strike on an aircraft based on maintenance technician logs. The precision and recall numbers not only demonstrates the effectiveness of the technique, but also shows the challenges involved in extracting information from cryptic free-form text contributing to the false negatives, and the drawbacks of a dictionary based approach with a number of false positives. The different data sizes used in the experimental study shows marginal effect on better precision and recall as the data size increases. The entity based analysis shows that, the technique is working better with high precision and recall numbers when there are relatively lower number of dictionary terms for entity extraction. In conclusion, we have shown an information extraction pipeline that can automate finding aircraft lightning strike related damages from unstructured text to improve SME productivity.

References

[1] G. Sweers, B. Birch and J. Gokcen, "Lightning Strikes: Protection, Inspection, and Repair,," Boeing, 2012. [Online]. Available: http://www.boeing.com/commercial/aeromagazine/articles/2012_q4/4/.

[2] R. C. Bunescu and R. J. Mooney, "A Shortest Path Dependency Kernel for Relation Extraction," in Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, 2005.

[3] S. Thenmalar, J. Balaji and T. V. Geetha, "Semi-supervised Bootstrapping approach for Named Entity Recognition," International Journal on Natural Language Computing, pp. 1-14, 2015.

[4] T. Yoshimasa and T. Jun'ichi , "Boosting precision and recall of dictionary-based protein name recognition," BioMed '03 Proceedings

of the ACL 2003 workshop on Natural language processing in biomedicine, vol. 13, pp. 41-48, 2003.

[5] A. Culotta and J. Sorensen, "Dependency Tree Kernels for Relation Extraction," in Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, 2004.

[6] H. Wachsmuth, "Text Analysis Pipelines," in Text Analysis Pipelines - Towards Ad-hoc Large-Scale Text Mining, Springer International Publishing, 2015, pp. 19-53.

[7] D. Jurafsky, "Spelling Correction and the Noisy Channel," [Online]. Available: https://web.stanford.edu/class/cs124/lec/spelling.pdf.

[8] S. Schuster and C. D. Manning, "Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks," 2016.

[9] D. Chen and C. D. Manning, "A Fast and Accurate Dependency Parser Using Neural Networks," in Proceedings of EMNLP, 2014.

[10] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861-874, 2006.