Information extraction from helicopter maintenance records as a springboard for the future of maintenance text analysis

Amber McKenzie[ac], Manton Matthews[a], Nicholas Goodman[bc], Abdel Bayoumi[bc]

[a]Department of Computer Science
[b]Department of Mechanical Engineering and [c]Condition-Based Maintenance Center
University of South Carolina, Columbia, South Carolina, U.S.A.

Abstract. This paper introduces a novel application of information extraction techniques to extract data from helicopter maintenance records to populate a database. The goals of the research are to preprocess the text-based data for further use in data mining efforts and to develop a system to provide a rough analysis of generic maintenance records to facilitate in the development of training corpora for use in machine-learning for more refined information extraction system design. The Natural Language Toolkit was used to implement partial parsing of text by way of hierarchical chunking of the text. The system was targeted towards inspection descriptions and succeeded in extracting the inspection code, description of the part/action, and date/time information with 80.7% recall and 89.9% precision.

Keywords: Natural language processing, Information extraction, Data preprocessing.

1 Introduction

Condition-based maintenance (CBM) is the practice of relying on mechanical indicators, such as vibration and oil debris, to detect and characterize faults and wear on machinery and basing maintenance practices on such indicators rather than on a given timeframe determined by the manufacturer-designated life of a part. CBM is a rapidly growing research field that relies mainly on vibration data and historical maintenance records to improve CBM techniques, identify new condition indicators, and refine algorithms. For several years, the Condition-Based Maintenance Center at the University of South Carolina (USC) has been involved with the South Carolina Army National Guard (SCARNG) and funded by the U.S. Department of Defense to conduct research to both enhance CBM practices and provide a cost benefit analysis. The data provided includes vibration data from fleet helicopters and historical maintenance reports which include maintenance test flight, part requisition, and fault and action reports. While much research and data analysis has been conducted on vibration data and select fields in helicopter maintenance records in order to improve and perfect CBM techniques, the wealth of information contained in the text-based portions of maintenance records (e.g. fault and action descriptions from TAMMS-A 2408-13-1 and -2 Army maintenance reports) has largely gone untapped due to the significant amounts of time and human effort required to analyze them by hand.

Around 2000, SCARNG migrated to an electronic system for maintenance record-keeping from their previous paper-based system. Though the records are now stored in a database, the

same basic record format was retained, including the text-based fields. This text would be much more suitable for the data mining necessary for CBM research if important key pieces of information were extracted out to populate queriable database fields. The value of this data for CBM research is significant and motivates this research to develop an efficient method for analyzing the data. Advances in the computer science/linguistics field of natural language processing (NLP), specifically information extraction (IE), have made it possible to employ IE techniques to allow a computer to extract usable information from these text fields in a form suitable to populate fields in a database, thus significantly augmenting the collection of CBM data available for computer-based analysis and facilitating the use of data mining techniques. Another goal of the research efforts at the CBM Center is data fusion, or to detect patterns and draw conclusions based on the integration of the different types of CBM data available.

2 Information Extraction

While many IE systems have already been developed for use in analyzing a variety of different types of texts, the majority of these systems are developed to analyze documents written in Standard English, such as news articles and literature. These systems are not suitable for use with this data, given the informal and often ambiguous nature of the language used in these reports. A tailored system must be developed that takes into account the unique and specialized vocabulary and syntax used by maintenance personnel in these reports, which is more similar to shorthand notation than Standard English usage. Research into information extraction in shorthand-type text for specialized domains mainly focuses around medical records [1-4]. Specific to a data set with a shorthand-type notation, Kraus et al. did work on extracting drugs and dosages from medical notes written by a physician [1]. For their approach, they used both supervised learning and a knowledge-based approach, which were well suited for implementation in their domain because of the consistent and predictable nature of the information they were trying to extract (i.e. drug names and numbers related to dosages). Also, a significant amount of work has been done on extracting temporal information from shorthand triage notes, medical records, and e-mails [3-5]. Little research has been done on information extraction for the maintenance record domain.

For the majority of these systems, a training set and gold standard dataset were necessary to train a system for use on the specialized set of data. In a domain where no such training set is available, much time and effort would have to be expended to produce a dataset sufficiently large enough to adequately train such as system. In addition, there is much room for human error during a supervised learning process that will result in more time to find and fix these problems later [6]. For this reason, basic NLP techniques were used to develop a new system capable of a basic analysis of helicopter maintenance records. It is the goal of this research not only to extract usable information to populate a research database, but also to utilize the system to assist in the creation of a sufficiently large tagged corpus that could be used for future research into automated information extraction from maintenance records in general.

2.1 Data

The dataset for text-analysis comprises historical TAMMS-A 2408-13-1 and -2 Army helicopter maintenance records, which contain information regarding reported faults or scheduled inspections/checks and the corresponding maintenance actions. Specifically, the text field in the

report containing the fault description or a description of the inspection or check performed is being targeted for the IE process. Certain notable characteristics of the dataset are easily computed [7]. It contains approximately 100,000 individual records with a lexicon of over 34,000 words or acronyms. The most interesting characteristic is that 80% of the records can be expressed using approximately 20% of the lexicon.

A number of challenges for the data processing system were identified after an initial examination of the text. In reference to the lexicon, the majority of the words are domain-specific, particularly regarding helicopter parts and maintenance actions, and consist of abbreviations or acronyms, some of which are specific to individual, or a group of specific, maintainers. Acronyms are not always consistent, and abbreviations are inconsistently marked with periods. Most of the fault descriptions do not constitute complete English sentences, and most are ungrammatical. Apostrophes are used to contract words and are occasionally omitted. Spaces are often omitted, and misspellings are common. Many decisions had to be made about how many and which dataset problems to resolve and which were not critical detriments to system performance.

Three main types of fault descriptions are able to be differentiated: inspections, checks, and fault/actions. Because of the highly variable nature of these reports, inspection descriptions were chosen as input for the system based on the fact that they are easily distinguishable from other types of descriptions, and the system output is more easily validated by cross checking with inspection code information. Inspection descriptions comprise roughly forty percent of the records in the entire dataset and contain information regarding the inspection code, a description of the inspection or the part involved, and when the inspection was due or completed.

2.2 Tools and techniques

The IE domain encompasses a broad range of researched techniques, as well as open-source toolkits, available for implementation in the development of an IE system. For this research, tools and techniques were used which were flexible enough to be tailored to suit the demands and requirements for analyzing the given text. The Natural Language Toolkit (NLTK) is an open-source package of programs and modules geared towards language analysis and computational linguistic tasks [8]. The advantages of this over other toolkits or IE systems are that it is open-source, provides basic, easy-to-use NLP functionality, and is completely self-contained [9]. The available programs and functions are able to be fine-tuned and manipulated to tailor them for specific needs. The Python programming language it is written in is well-suited for NLP tasks [9]. In this way, NLTK was seamlessly applied to developing an IE system from the bottom up and was adaptable enough so as not to constrain creative development.

A partial parsing approach was implemented because a full text analysis was not necessary in order to extract the desired information. Abney's technique of finite-state cascades was implemented to handle the nested syntactic structure needed to capture meaningful chunks of data from the text descriptions [10]. These cascades obviate the need for a complete grammar of ungrammatical input and allow a parsing of only relevant, desired information from the text.

3 Methodology

For this research, a sequence of NLP tasks were individually tailored to process text for this specific engineering domain. Given the wide range of NLP and IE tools available, it was

necessary to begin with basic, individual NLP processing techniques in order to facilitate the identification of potentially useful tools that would be uniquely applicable to the maintenance domain. An individually tailored IE system will also enable the faster processing of text to develop a gold standard and training corpus with which to implement machine learning techniques to discover extraction patterns, templates and rules, thus allowing for the automation of IE system development in the future for records in the same domain. The developed IE process involves four main steps: text preprocessing, part-of-speech (POS) tagging, chunking/parsing, and relation extraction. These steps combine to produce a template that is filled out for each record and is easily migrated into the database. Templates also play a critical role in allowing for the evaluation of the system [11]. The main goal of the system is to produce extracted information that, once entered into the database, will greatly facilitate data mining CBM research and data fusion efforts.

3.1 Text Preprocessing

The historical helicopter maintenance data is stored in a database and currently accessed through Microsoft Access, though database renovations and improvements are currently being implemented. The final form and interface for the database have yet to be determined. For this reason and for the ease of system development, the relevant text fields were simply exported as a text document for system input. In the future, a process will be put in place to automatically extract the relevant data for processing, including data from other fields in the records which are currently not integrated in the text-analysis process.

Once the target data is saved into a text file, it must be preprocessed before being considered for tagging. All of the reports are recorded with all of the text in capital letters, so case differentiation does not factor into the analysis. Each line of text represents one fault description and is first split by white space. Then each token is processed individually. It is broken around symbols (except '/', which is left in the middle of a token to handle instances such as "W/IN"), with periods between or before numbers left attached to that token. Also, apostrophes before "S", "D", and "T" remain in the token. This handles cases such as "REQ'D", "PILOT'S", and "WON'T". Other apostrophes are used as quotation marks and should be broken from the token. The output of this process is a list of tokens for each line, and thus a list of lists.

3.2 Part-of-speech Tagging

The nonstandard and often ambiguous nature of the maintenance descriptions posed significant challenges for POS tagging. To address this challenge, several different types of taggers were trained and implemented in a 'back-off' system, in which any word not tagged by a given tagger will be sent to the back-off tagger, as illustrated in Figure 1. Any words that are not tagged by any of the taggers are ultimately tagged with the default tag. In this case, the default tag is NN (noun), as nouns are the most common part of speech for the dataset. For this tagging process, the tag set used is a modified version of the Penn Treebank Tag Set [12]. Several tags were added to tag specific, high-frequency, semantically-significant words, e.g. INSP for any form of inspection and DUE for "due".

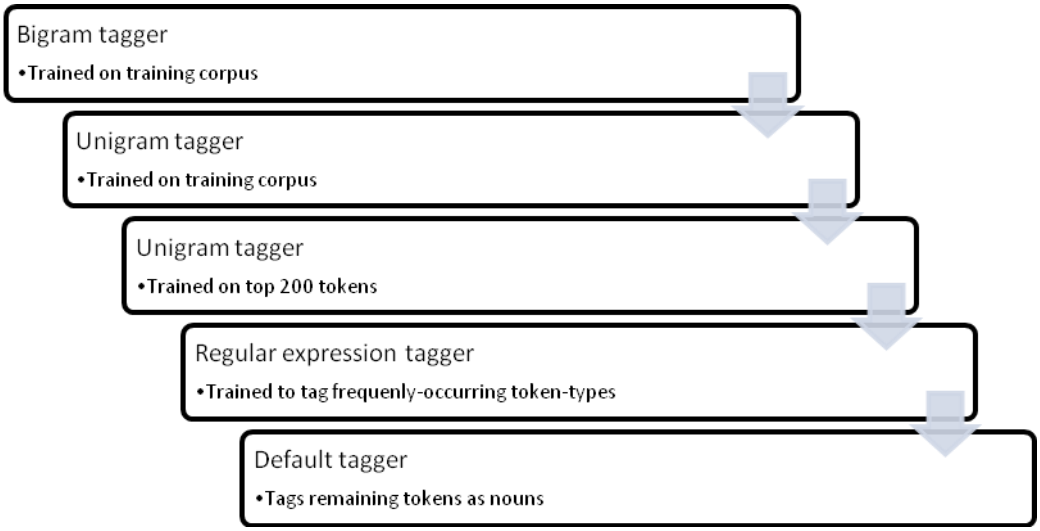

Figure 1 – Progression of POS tagging

A small training set of approximately 175 descriptions was manually tagged and used to train both a bigram and unigram tagger. The training set consisted of a mixture of different description types to ensure that the system was not tailored for one specific type. The bigram tagger achieved limited effectiveness given the limited training set it was provided. The training set was also used to train a Brill tagger, which did not perform as well as the step-down tagging system and was therefore not used as part of the IE program. (It is likely that both the bigram and Brill taggers would achieve significantly improved performance if trained on a much larger tagged corpus.) Those tokens not tagged with the bigram tagger are run through the unigram tagger trained on the same training set. The back-off tagger for this unigram tagger is another unigram tagger trained on a tagged set of the top 200 most frequently occurring tokens in the dataset. Because the training set and top 200 words comprise a relatively small portion of the lexicon, a regular expression tagger was installed to catch frequently-occurring types of tokens. Regular expressions were written to catch the following tokens:

- CODE: a sequence of a letter followed by numbers
- D: an eight-digit date, e.g. 20040327
- CD: all types of number formats
- LE: a single letter
- SYM: symbol other than '/', '#' or '.'
- VBN: verbs ending in '-ED'
- VBG: verbs ending in '-ING'
- CK: the word 'check' or any variation/abbreviation of it

The process of tagger development involved much trial and error, as is the case in any system development. Frequently-occurring or semantically-significant words which were incorrectly tagged were simply added to the tagged set of top words. For future research as the system is further refined, tests will be run to determine if adding a few extra POS tags will be sufficient to enhance the performance of the chunker or if it is necessary to examine the word content in order to correctly identify more detailed semantic chunks.

3.3 Chunking

Because of the specialized nature of the text domain, a layered regular expression chunking technique was chosen to provide a partial parse of the text-based maintenance entries. Again, trial and error was necessary to determine the effectiveness and success of chunking certain tag patterns. The inspection descriptions generally contain three main parts: the inspection code, a description of the inspection action or the part inspected, and some form of time indication as to when the inspection was due or took place (either date or hours or both). For this research, these were the goal chunks to be extracted. The inspection code and time information easily translates into database entries. The description text is left in its original form to ensure the generic nature of the system and make it usable for a variety of maintenance records. Future research is geared towards incorporating an ontology and knowledge base that are domain specific and will allow for the extraction of parts and actions specific to helicopter maintenance. The generic nature of the program promotes a broad application of the system in a variety of fields and provides the capability of making the system more domain-specific with the use of a domain-oriented ontology.

For the regular expression chunker, the following chunks were specified for identification:

- DATE: includes many forms that a date appears as
- CODES: includes both words tagged as codes and codes consisting of certain sequences of tags
- INSPECT: different syntactic formats that denote an inspection
- RANGE: text specifying a range of numbers
- HRS: text specifying a specific time or range of hours
- VP: any of a variety of verbs or verb phrases
- DUEPP: denotes a phrase identifying when something is due
- PP: prepositional phrase
- NNP: noun phrase
- PAREN: set of parentheses and the text inside those parentheses
- TIME: date or hours
- DESC: part of record detailing the inspection

The regular expressions written for these chunks are ordered specifically to allow the inclusion of smaller chunks in the specifications for later chunks, thus resulting in the cascading effect. Parentheses were excluded (chinked, rather than chunked) and were not considered to be part of any other chunks. Generally, information included in parentheses is not significant and need not be considered when analyzing content. While this list represents the majority of chunks needed for IE from all types of descriptions, some other chunks were included, which are necessary to adequately analyze text descriptions other than for just inspections.

3.4 Relation extraction

Once the desired information has been chunked, relations between the entities are established to provide some context for the extracted information. For inspection descriptions, this process is rather basic and almost unnecessary, but it is modeled in the system because it will become necessary with the inclusion of other description types. For inspection descriptions, relationships exist among all three of the extracted parts of the description. Thus, three relation triples were formed:

- ([<CODE>], ‘at’, [<TIME>]) : code occurred at given time
- ([<DESC>], ‘at’, [<TIME>]) : action/part inspection occurred at given time
- ([<DESC>], ‘describes’, [<CODE>]) : action/part description describes given code

These relations represent the first step in the process of analyzing the semantic structure of the records in order to extract not only flat pieces of information, but also information that is meaningful and content related.

4 Results

4.1 Template

For this research, the IE process was based on an extraction template that represents the information that is targeted for extraction from an individual inspection description. The template consists of four fields: code, action/part, date, and time (as represented in the bottom portion of Figure 2). While this is the information generally included in an inspection description, all four pieces of information are not always included for a given report. Some records do not contain a code; others may contain a date or a time, or may lack a timestamp altogether. Not only was this template critical to determine what information was to be extracted, but it is also used to structure the extracted information for insertion into the database.

Sample inspection description:

INSP A822-(PMS) INSP FOR APU ANTI-FLAIL AND #7 DRIVESHDUE AT: 20031105 OR 1590.0 HOURS FOR RHA

Code:	A822
Action/Part:	APU ANTI - FLAIL
Action/Part:	# 7 DRIVESH
Date:	20031105
Time:	1590.0

Figure 2 – Extraction template for inspection descriptions

4.2 Performance

The assessment of the performance of the system was broken into stages, and the POS tagger and chunker/parser were analyzed separately. The POS tagger achieved a correct tagging rate of 93.29% when run on a test set of 64 individual maintenance descriptions, which included a variety of description types. Table 1 presents a confusion matrix representing the types of mistakes the tagger made. Due to space considerations based upon the number of POS tags used, the matrix only includes the parts of the matrix that represent errors. Tags for which all instances in the test set were tagged correctly, and which were therefore not represented in the matrix, include: CC, CD, CODE, D (date), DASH (hyphen), DUE, HR, MD, NP, PAR (parentheses), PER (period), SYM, TO, and WRB. When provided with a test set comprising only inspection descriptions, the tagger achieved a performance of 96.20%, which gives evidence to the more predictable nature of these record types as compared with other types. The largest numbers of

errors for both test sets occurred when a token was not tagged by any of the taggers and was given an incorrect NN tag. Many of these incorrectly-tagged tokens represent misspellings and words that require more refined preprocessing.

Table 1 – Confusion matrix for POS tagger, displaying only those categories where errors occurred.

	CK	DT	IN	INSP	JJ	LE	NN	NUM	RB	VB	VBD	VBG	VBN	VBP	VBZ
CK	[9]						2								
DT		[3]					1								
IN			[66]												
INSP				[19]			2								
JJ			1		[35]		17								
LE						[5]	1								
NN					1		[254]								
NUM								[4]							
RB					1		2	1	[6]						
VB				1			3			[5]					
VBD							1				[2]		3		
VBG							2					[4]			
VBN							1						[12]		
VBP							1						1	[1]	
VBZ							4								[2]

(rows: reference; columns: test)

For comparison, the gold standard was altered to include only tags from the Penn Treebank tagset and was compared to the output of the NLTK's currently recommended pos tagger, which is a classifier-based MaxEnt tagger [13]. This type of tagger does not require training data, but rather relies on a prebuilt, feature classification model to determine part of speech of a given word [14]. The MaxEnt tagger recorded a performance of 73.18%, which gives credence to the assertion of performance improvement achieved by the developed pos-tagging system. For further proof of improvement, the test set was run through both a unigram and bigram tagger trained on the training set. By themselves, without the support of the other parts of the system, the unigram and bigram taggers achieved accuracy of 48.71% and 13.14%, respectively, which represents a significant decrease in performance from the hybrid system. The extremely poor performance of the bigram tagger is due to the small size of the training set, and the fact that it did not have a backoff tagger to resort to when it could not determine the pos of a certain word.

The chunker was evaluated using the standard metrics of precision, recall, and F-measure. Each text description in the test set was manually reviewed to determine how many chunks were identified by the system, how many actual chunks were in the record, and how many chunks were correctly identified. The system achieved a precision of 89.93% and a recall of 80.72%. The F-measure was 85.08%. These numbers demonstrate that the chunker was rather successful at identifying information needed to be extracted, but also that there remains room for improvement using other NLP techniques.

The success of the chunking technique chosen for use in this work is wholly dependent on the chunking patterns built into the system. Because of this dependency, it is not possible to compare it to another similar system without simply providing a different set of extraction patterns for the same chunker. However, this comparison would be unrealistic, as these patterns would not represent an efficient parse of the input data and would certainly perform poorly.

Once other chunking systems are implemented, a comparison can be made between the different parsing techniques, but this implementation is a task that is scheduled for future work on this system.

5 Conclusions and Future Work

A critical need was identified in the engineering maintenance research domain for a system to reliably analyze text-based maintenance records in order to populate a database to facilitate data mining research efforts. To address this need, an IE tool was developed to extract meaningful chunks of information and identify the relations between extracted entities. Specifically, this research targets Army helicopter maintenance records, particularly inspection text descriptions, to support research efforts to improve CBM techniques and practices. The developed IE system was successful at identifying the inspection code, description, and time and/or date information from individual descriptions. Further research will focus on refining the system to handle other types of maintenance action descriptions, which contain more variability and ambiguity. Also, if data from other fields in the records is incorporated, the validity of the system could be more firmly established, and that data might be able to be incorporated in the IE process to improve performance.

The IE system described in this paper is just the beginning step in an endeavor to provide a reliable system for analyzing text-based engineering-domain maintenance records. A variety of future research tasks are anticipated to address specific needs of such a system. The system described in this paper will be used to tag a much larger training corpus of records to greatly reduce the manual labor required to produce such a dataset. This corpus will be used to train both Brill and bigram (and possibly trigram) POS taggers to determine if these types of taggers could prove to be more effective and efficient. A larger available training corpus will also facilitate the use of machine learning techniques to determine more detailed and refined extraction patterns. The inclusion of a domain-specific knowledge base/ontology will provide semantic content identification capabilities for the system, which, in turn, will allow for more in-depth analysis of the maintenance record content.

References

[1] Kraus, S., Blake, C., & West, S. L. (in press). Information extraction from medical notes. *Proceedings of the 12th World Congress on Health (Medical) Informatics – Building Sustainable Health Systems (MedInfo),* Brisbane, Australia, 1662-1664.

[2] Hripcsak, G., Griedman, C., Alderson, P. O., DuMouchel, W., Johnson, S. B., & Clayton, P. D. (1995). Unlocking clinical data from narrative reports: A study of natural language processing. *Annals of Internal Medicine,* 122, 681-688.

[3] Gaizauskas, R., Harkema, H., Hepple, M., & Setzer, A. (2006). Task-oriented extraction of temporal information: The case of clinical narratives. *Thirteenth International Symposium on Termporal Representation and Reasoning* (TIME'06), 188-195.

[4] Irvine, A. K. (2008). Natural language processing and temporal information extraction in emergency department triage notes. *A Master's Paper.*

[5] Han, B., Gates, D. & Levin, L. (2006). Understanding temporal expressions in emails. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, 136-143.

[6] Cardie, C. (1997). Empirical methods in information extraction. *AI Magazine*, 18(4), 65-79.

[7] Bayoumi, A., Goodman, N., Shah, R., Eisner, L., Grant, L., & Keller, J. (2008) Conditioned-based maintenance at USC – part II: Implementation of CBM through the application of data source integration. Presented at *the American Helicopter Society Specialists' Meeting on Condition Based Maintenance.* Huntsville, AL.

[8] Bird, S. G. and Loper, E. (2004). NLTK: The Natural Language Toolkit. *Proceedings, 42nd Meeting of the Association for Computational Linguistics (Demonstration Track)*, Barcelona, Spain.

[9] Madnani, N. (2007). Getting started on natural language processing with Python. *ACM Crossroads*, 13(4).

[10] Abney, S. (1996). Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4), 337-344.

[11] Cowie, J., & Lehnert, W. (1996). Information extraction. *Communications of the ACM*, 39(1), 80-91.

[12] Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2), 313-330.

[13] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: analyzing text with the Natural Language Toolkit*. O'Reilly Media: Sebastopol, CA.

[14] Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition.* Pearson Education, Inc.: Upper Saddle River, NJ.