Automated Data Contextualisation for Decision Support

Thomas McKeon

Tim French
School of Computer Science & Software Engineering

Melinda Hodkiewicz
School of Mechanical & Chemical Engineering

Wei Liu
School of Computer Science & Software Engineering

Abstract

This paper describes a method to create fast, accurate and unbiased textual summaries in a natural language from statistical data using text mining and Natural Language Processing (NLP) techniques. The system is tested on data from a despatching industry. The textual data found in niche industries is generally unsuitable for NLP as the text is unstructured, irregular and domain specific. This paper demonstrates how supervised automatic text processing can result in machine-interpretable text with an 81.5% reduction in out-of-vocabulary text. The system then uses the cleaned text to infer rules between historical reports and other streams of operational data. These rules are used to generate textual summaries automatically. The use of this automated system can assist decision-making for businesses by providing role-specific, real-time reports for managers and reducing the cognitive load for individuals required to produce these reports.

1. Introduction

Despatch operators such as air traffic controllers and plant operators often sit in front of multiple computer screens and are required to synthesise many streams of data. This can be composed of tabular information, graphs and animated mimics of site operations. From these multiple streams of information they form a mental model of how the site is operating. When events cause deviations from the intended schedule they use their experience to deduce cause-and-effect relationships and make decisions to return to normal operation. At the end of each day the operators are required to summarise the key events as part of their routine reports. What is or is not considered as important to disclose is a decision made by the operator and may be affected by events and the consequence of decisions made during the shift. This can lead to unconscious bias in reporting which is difficult to detect by management who have not been involved in the detailed operation. Ideally, data from the shift would be used to identify the main causes of unplanned events and this is what would be reported, with accompanying text describing the cause and effect(s).

The paper demonstrates a process for creating a system to generate automated reports in a natural language based on a set of statistical inputs. The system has tested a despatching industry which has a niche lexicon, typical of most industries. Most of the available textual information is short, consisting of less than 12 words and have limited use of linguistic rules. The texts are noisy containing amounts of erroneous or irrelevant data and contain many instances of jargon, abbreviations, acronyms and identifiers, specific to the the domain.

2. System Architecture and Components

Figure 1 depicts the system architecture, and is comprised of three main modules: Text Processing, Template Generation and Language Generation. There is no well known framework for an automated report generation system due to the adaptations required for different applications and the limited textual information available for niche industries. However, there are existing methodologies that have been used to perform typical text processing techniques and Information Extraction (IE), further explained in Sections 2.1, 2.2 and 2.3.

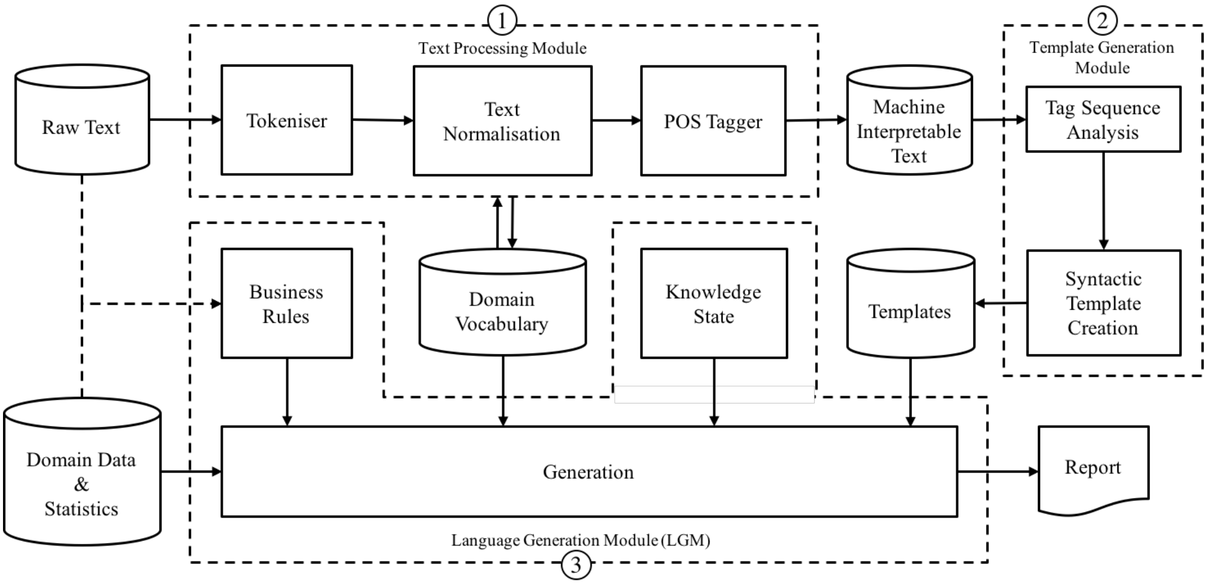

Figure 1 The system flow including the architecture for the Text Processing, Template Generation and Language Generation Modules

2.1 Text Processing Module (1)

Company textual data, such as text found in shift reports is generally short and often does not conform to rules of spelling, grammar and punctuation. Text normalization is commonly seen as cumbersome and the system demands high accuracy for acceptable use of data in later applications. The text processing module has two applications: for pre-processing on the unclean, unstructured text transforming it into a clean machine-interpretable form for use in automated Natural Language Processing (NLP) tasks; and as a method for extracting domain specific vocabulary, based on the work by Pennell & Liu (2014), Li & Liu (2015) and Sproat et al. (2001). The text processing module extracts tags each word with a Part-of-Speech (POS) tag and is a critical part of this NLP system. POS tagging is the process of word-category disambiguation, assigning a label to each token, such as ‘ADJ’ for an adjective or ‘NUM’ for a cardinal number. The tags are an extension of the universal tagset developed by Petrov et al. (2011) as demonstrated in Figure 2a in Section 3.1. Evaluation of accuracy is determined by comparing automated annotated to manually annotated data.

2.2 Template Generation Module (2)

Syntactic templates, similar to those used by Theune et al. (2001) are created through analyzing the patterns of tokens, the ordering of POS tags and the linkages between operational parameters such as event types and instances of equipment (Table 2a). This results in tree-like syntactic structures which each have associated metadata based on frequency pattern mining.

2.3 Language Generation Module (3)

Language generation is used to produce a natural language text expressing the system's input data. The input data comprises of a number of simple statistics and known instances (e.g. 'Car1'). The Language Generation Module (LGM) performs tasks including: content determination to decide what information should be expressed; discourse planning to determine which template to select given the current knowledge state; and sentence aggregation and lexicalization to ensure that the right words are selected and are used in the correct order to make linguistic sense. The Knowledge State selects a template given the domain data (topic) and if its conditions evaluate to true, given the input data (Figure 3a). Initially all input fields are labelled as 'unknown'. As fields are filled iteratively, where the search space of possible slot fillers for a given gap are reduced, given the previously chosen inputs in accordance with business and language structure rules. For example, specific equipment combinations can never exist in the same phrase or certain actions can only be performed on specific types of equipment.

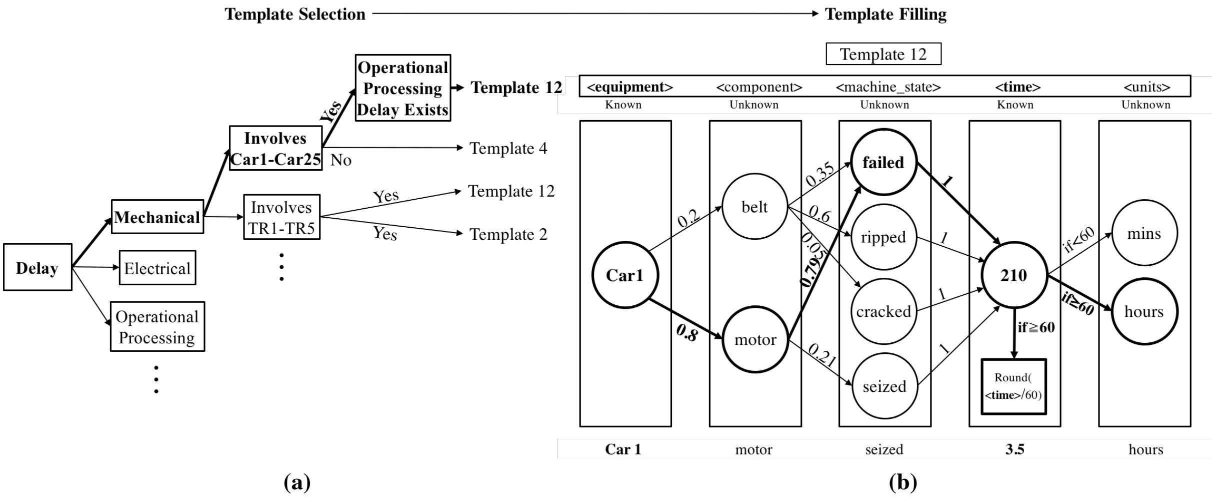

Figure 2 (a) A template can be used if it belongs to a topic of the current data and satifies the conditions of the current Knowledge State
(b) Example of how a syntactic template is filled during the generation of the sentence: *Car1 motor failed 3.5 hours*

3. Results and Discussion

3.1 Text Processing Module (1)

Short messages, especially those contained to a maximum length and those within a niche domain contain deviations from the standard English language. The unclean text considerably hampers the accuracy of the subsequent modules within the system due to the large amount of out-of-vocabulary (OOV) words which can be incorrectly interpreted. Effort has been made to detect and correct errors as the amount of noise in the data puts an upper limit on the accuracy of the system.

The text is broken up into whitespace-separated tokens. Initially, tokens are marked as OOV using simple dictionary criterion, i.e. if the token is not in the standard English dictionary it is marked as OOV. This results in the raw text corpus with 31.9% OOV after tokenisation. The highest sources of OOV tokens were found to be due to: spelling mistakes; incorrect variations of abbreviations, identifiers, acronyms and entity names; and missing spaces causing incorrect word segmentation. Each of the Text Normalisation subtasks conducts ‘cleaning’ of OOV words and each of the noises fall into the categories listed in Table 1. The table also shows the reduction in percentage OOV after each operation is performed and demonstrates the effect of each subtask on a given sentence.

Level	Task	% OOV	**Original Text:** ‘Today RW2& rw% blckd due 2 dmgd plane (12hr).’
Word	*Case Normalisation*	31.9	‘today rw2& rw% blckd due 2 dmgd plane (12hr).’
	Token Segmentation	28.6	‘today rw2 & rw% blckd due 2 dmgd plane (12hr).’
	Punctuation mark Correction	28.2	‘today rw2 and rw5 blckd due 2 dmgd plane for 12hr).’
	Misspelled word correction	16.1	‘today RW2 and RW5 blocked due to damaged plane for 12 hrs).’
Sentence	*Case restoration*	16.1	‘Today RW2 and RW5 blocked due to damaged plane for 12 hrs).’
	Remove select punctuation	12.4	‘Today RW2 and RW5 blocked due to damaged plane for 12 hrs .’
	Extra space deletion	5.9	‘Today RW2 and RW5 blocked due to damaged plane for 12 hrs .’

Clean Text: ‘Today RW2 and RW5 blocked due to damaged plane for 12 hrs.’

Table 1 Text Normalisation Subtasks and corresponding (highlighted) effects on an example text from the original text resulting in clean text.

The output from the Normalisation process reduces the OOV text by 81.5% from the original 31.9%. The majority of the remaining 5.9% OOV text is due to inadequate correction of abbreviations since many are not captured during Named Entity Recognition (NER), creating an incomplete dictionary. Abbreviations are not easily extracted and recognized using a set of rules due to the uncommon variations and limited historical usage, such as for the term ‘change out’ having in-text representations including {‘C/Out’, ‘C/O’, ‘Chg/Out’, ‘C.out’, ‘COut’}. Since the Domain Vocabulary (explained in Section 3.1.1) is insufficient the Text Normalisation process cannot remove all OOV words. The rest of the noisy entities, acronyms and asset identifiers can be cleaned by replacing all OOV words with the standardised form using a set of rules and a lookup table, explained further in Section 3.1.1. The Aspell spelling correction algorithm corrects all other OOV words.

POS-tagging specifies the semantic class of each word using the extended universal POS-tags An example of POS tagging is shown in Table 2a, using an adapted version of the Stanford Tagger (Toutanova & Manning, 2000).

Sentence:	Car1	motor	failed	3.5	hours
Universal:	ID	NOUN	ADJ	NUM	NOUN

Table 2 Example English sentence and corresponding POS tags

3.1.1 Domain Vocabulary

The domain vocabulary is formed in parallel to the Text Normalisation process and is also used within the subtasks listed in Table 1. The vocabulary is composed of four libraries: an Asset ID List, a Noun List, an Abbreviation List and an Acronym List. These are formed using a set of regular expression rules (Figure 3) and once complete, used to convert all identified tokens in the raw corpus to the *standard* forms. For example, each library contains a standard form such as ‘C1’ and its multiple variants, ‘Car 1’ or ‘car1’ or ‘C!’. The standard form is chosen based on the most frequent instance in the corpus. However, the expanded form such as ‘Car 1’ is usually manually assigned. The vocabulary contains 345 Asset IDs, 41 Acronyms and 736 Nouns, each having a set of variants.

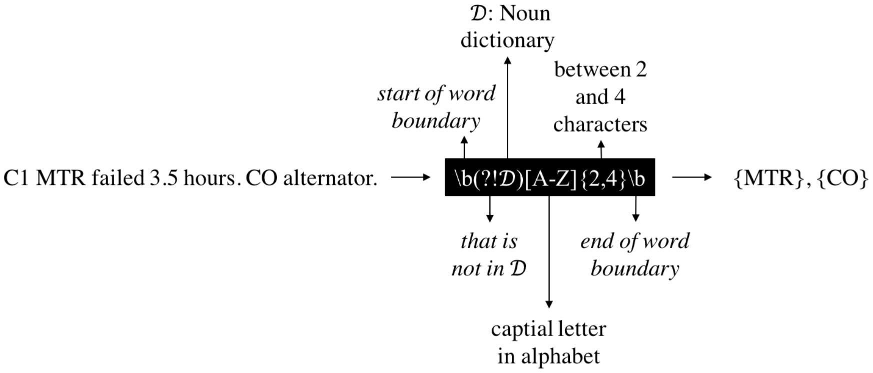

Figure 3 Example of extracting typical acronyms using a regular expression where 𝓓 is the concatenated list of all known nouns.

3.2 Template Generation Module (2)

The syntactic templates $\psi = \langle S|E|C|T \rangle$ are inferred by common sequences of semantic classes of the words found within the corpus. These parsed sentences form tree-like structures as seen in Figure 4a, where S is the syntax tree, E is a set of linked data to be substituted into the input fields in S, C is the conditions of applicability of the elements in E given the Knowledge State T. The templates are assigned metadata comprising of a list of associated popular event types (e.g. Unscheduled Electrical) based on historical frequency of occurrence and other information such as the total set of equipment types that have ever existed in a given sequence. Template generation is an ongoing procedure and will be made fully automatic from the manual process that is used currently. The templates have a sentence structure akin to the Attempto Control English (ACE) framework, however this controlled natural language is too restricted for the language use in the domain and must be modified for use.

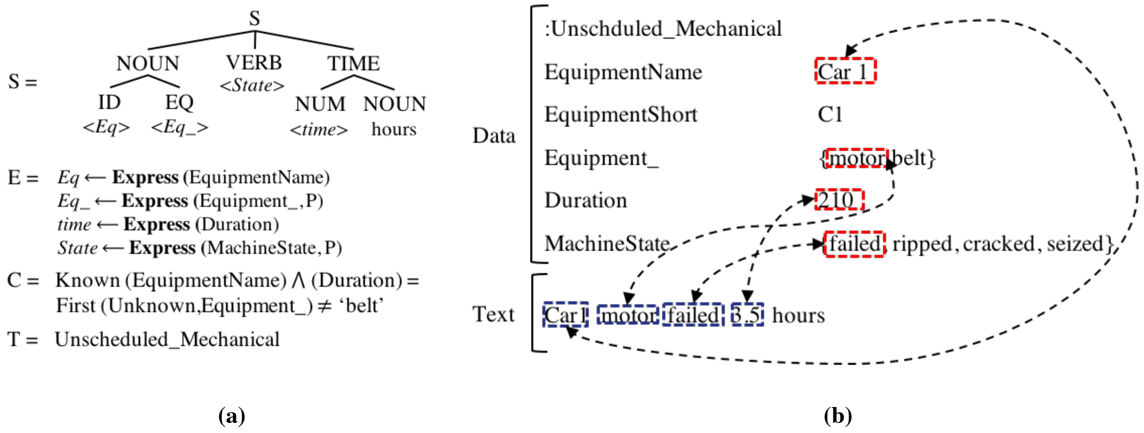
S =
S
NOUN
VERB
TIME
<State>
ID
EQ
NUM
NOUN
<Eq>
<Eq_>
<time>
hours
E = Eq ⟵ Express (EquipmentName)
Eq_ ⟵ Express (Equipment_, P)
time ⟵ Express (Duration)
State ⟵ Express (MachineState, P)
C = Known (EquipmentName) ∧ (Duration) = First (Unknown,Equipment_) ≠ 'belt'
T = Unscheduled_Mechanical
Data
:Unschduled_Mechanical
EquipmentName
Car 1
EquipmentShort
C1
Equipment_
{motor, belt}
Duration
210
MachineState
{failed, ripped, cracked, seized}
Text
Car 1 motor failed 3.5 hours
(a)
(b)

Figure 4 (a) Sample syntactic template ψ (b) Aligning text and data

3.3 Language Generation Module (3)

The automatic generation of reports requires adequate statistical information to determine the Knowledge State and ensure an appropriate template is chosen to satisfactorily describe the situation. The linking of the templates and the statistics (Figure 4b) is manually determined and the template filling is done semi-automatically. Input fields that are unknown in the chosen template are inferred by determining the likelihood of a word occurring given that a word has been selected either before, after, or before and after the current field using Bayes' Theorem. The process for language generation will be made fully automatic as part of the project.

4. Conclusions and Future Work

An automated report generation system has multiple benefits that can add value for both employees and for a business in general. An automated system can deliver live information in a human-understandable form on a current situation and deliver it to specific people, in the correct persona. For example, the generated text could include expanded acronyms depending on their tacit knowledge and business understanding. Providing a description of the situation with white-box reasoning can reduce the cognitive load on an individual removing the need to piece together multiple forms of information. The system can also deliver unbiased information since it is based on fact, providing management with a clear picture of the actual occurrences.

The system has reduced the number of OOV words in the original text data by 81.5%, producing a structure that is machine-understandable. Further work involves measuring the performance of the proposed system in various domains and developing the functionality of the modules presented in Figure 1. Improvement in the algorithms for spelling correction and abbreviation detection will aim to reduce the percentage of OOV text to nil. Development of the POS tagger aims to allocate the correct tag with near to 100% accuracy. The Template Generation and Language Generation Modules will be automated and evaluation of the generated reports will be conducted. This will be determined by comparing results with human generated reports over various measures. The improvement of the business rule base and business statistics with the addition of more complex templates will enable more information to be expressed about a specific event, such as including the cause(s).

6. References

Duma, D. and Klein, E., 2013. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013),* pp. 83-94.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pp. 63-70.

Li, C. and Liu, Y., 2015, June. Joint POS tagging and text normalization for informal text. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, pp. 1263-1269.

Pennell, D.L. and Liu, Y., 2014. Normalization of informal text. *Computer Speech & Language*, ***28***(1), pp. 256-277.

Petrov, S., Das, D. and McDonald, R., 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Sproat, R., Black, A.W., Chen, S., Kumar, S., Ostendorf, M. and Richards, C., 2001. Normalization of non-standard words. *Computer Speech & Language*, ***15***(3), pp. 287-333.

Theune, M., Klabbers, E., de Pijper, J.R., Krahmer, E. and Odijk, J., 2001. From data to speech: a general approach. *Natural Language Engineering*, ***7***(01), pp.47-86.