INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

IDENTIFYING DESIGN ERRORS AND HUMAN ERRORS USING AUTOMATIC CLASSIFICATIONS

Sanghee Kim[1], and Ken M. Wallace[1]

[1]Engineering Design Centre, Engineering Department, University of Cambridge, U.K.

ABSTRACT

Current information retrieval systems are mainly designed to retrieve a list of documents. Given a short query, the retrieval systems are able to locate relevant documents using a variety of content matching methods. The need to classify and retrieve sentences instead of whole documents therefore poses a new challenge. In accident reports, for example, it is important to identify and distinguish causes of accidents arising from Human Errors and Design Errors. The automatic identification of sentences describing Human or Design would be feasible if sentence-level classifications were possible. Human Errors, e.g. *the pilot forgot that a conditional crossing clearance was still pending,* are known to cause up to 70% of aviation accidents or incidents, and are therefore a major concern of the aviation industry. However, it is important to not just focus on the errors made, but also to comprehend the underlying misconceptions that lead to them, especially the potential influence of aircraft designs on the behaviour of human operators. Previous research developed a sentence classification that simply differentiated Human Errors from Design Errors. Human Errors can be described by various factors. In order to focus on recurrent and critical ones, a detailed categorisation of such factors is necessary. Such a categorisation will provide a better means of determining analytically common human factors across accidents and the level of detail necessary to apply the approach in industries. This paper presents the results of research to classify sentences into multiple categories and demonstrates that this leads to a better understanding of the accidents.

Keywords: Human and design errors, natural language processing, supervised learning approach

1 INTRODUCTION

With advances in modern design and technology, the number of critical aviation accidents has decreased over the past 50 years. However, the proportion of accidents caused by Human Error have not reduced, in fact the failures of human operators when dealing with the demands of complex systems, such as modern aircraft, have become the major cause of accidents. That is, accidents are caused by the features in the basic design that under normal operating conditions have no effect on operator performance. However, under certain circumstances, e.g. emergencies, these features lead to acute or chronic deterioration of operator performance causing an accident. Research has found that over 70% of aviation accidents can be attributed to Human Error [1]. However, in some cases, the real causes of such accidents can be attributed to the underlying design, e.g. *the pilot was overwhelmed with a high workload and stress due to the lack of a memory aid to record the conditional clearance.* By definition, Human Errors are inappropriate or undesirable human decisions or behaviours that reduce effectiveness, safety or system performance. Likewise, Design Errors arise because of incorrect knowledge and thus unintentionally deviate from a sound design. This can be because of poorly documented requirements or a misunderstanding of the totality of what the design must accomplish. Thus, in order to design safer systems, designers need to understand the role that automated functions and human-computer interfaces may have played in contributing to Human Errors that lead to incidents and accidents.

Accident reports are an essential resource to understand interaction failures between operators and systems. These are published as a result of an investigation carried out by a team of specialists after an accident happens. The facts, conditions, circumstances, and probable causes of accidents are described in the reports. When an investigation is completed, the report is put on public Web sites, which are usually maintained by governments. Previous research has investigated the limitations of using accident reports for identifying Human Errors induced by Design Errors [2, 3, 4, 5]. Two main reasons for the limitations were identified. The first reason is that current accident reports do not capture the information in a way that specifically informs future designers effectively about the causes of previous failures. For example, retrieval systems do not include a search option that classifies the reports contents according to the identified causes making it easy to extract the accidents caused by Human Errors. The second reason is that retrieval systems do not provide an efficient summary that highlights important information. It is therefore a time-consuming task to read all the reports since each report is generally around 2-3 pages and contains over 80 sentences. It is also possible that without systematic support, designers might draw false conclusion that have no directly links to the accidents. A computer-supported tool that automatically highlights the sentences related to Human Errors and Design Errors would therefore be very helpful.

Sentence-level classification poses a new challenge to traditional text classifications that are concerned with classifying a whole text into pre-defined categories. Cue phrases like *forgot* can be used as linguistic markers. For example, the sentence of *the pilot forgot that a conditional crossing clearance was still pending* should be classified into Human Error by locating *forgot*. Using such semi-fixed cue phrases can be useful for many applications such as text summarisations, semantic orientations and information extractions. However, since the descriptions of both types of Error are mostly implicit and ambiguous, it is difficult to create reliable and scalable cue phrases. Systems that use cue phrases usually rely on manually created lists, the acquisition of which is time-consuming, error-prone and difficult to transfer across applications. Attempts for automatic cue phrase identification have used a string-based pattern matching, but this has produced only limited performance.

In previous research, a supervised learning approach that used a wide variety of linguistic features was applied to classify the sentences into either Human Errors or Design Errors [6]. Tests showed an improved performance compared to the identification using cue phrases. However in order for the approach to be used in the aviation industry, an extension was necessary. In practice, once the accident investigators have identified human factors as causing an accident, a detailed analysis of the factors is carried out in order to separate safety-critical issues from minor ones. Based on this analysis, safety authorities determine which human factors are critical, recurrent and thus need an attention. It is believed that the causes of accidents do not happen in isolation; but that they are the result of a chain of active and latent causes each one affecting the next. In the literature, a hierarchical categorisation of Human Errors and Design Errors has been proposed in order to establish better any links to Design Errors that contribute to Human Errors. Since the previous research was only able to tell whether a given sentence was related to Human Error or Design Error, the investigators need to check the extracted sentence manually in order to identify the detailed types of human factor involved. It is therefore necessary to extend the research to provide the level of detail necessary for the approach to be more practicable. The research has therefore been extended to include multiclass classifications that separate sentences into one of three or more classes. This paper presents the results of this research and discusses how the extension helps to improve the identification and the understanding of the impact Design Errors have on human factors. The main objective of this research is to understand how the information extracted from accident reports can be made to be more accessible to engineering designers.

2 RELATED WORK

The classification of sentences according to their semantic types draws concepts from various research areas including Information Extraction (IE), cue phrase identification and sentence-level classifications. IE, which is a sub-field of Natural Language Processing (NLP) has been commonly used to extract domain entities of interest, e.g. person names or dates, from unstructured texts [7]. IE uses shallow NLP techniques, e.g. Part-Of-Speech (POS) taggings, and stores the extracted information in database-like structures. IE performs well in extracting domain entities using lexical-syntactic patterns, i.e. a person name is preceding a title (Mr) and starts with capital letter. However, IE is not suitable for this research since the descriptions of Human Errors and Design Errors are often implicitly expressed making it difficult to create the extraction rules. In addition, traditional IE systems aim to extract all the entities as completely as possible, whereas the proposed approach identifies a small number of sentences that are deemed as related to either Human Errors or Design Errors.

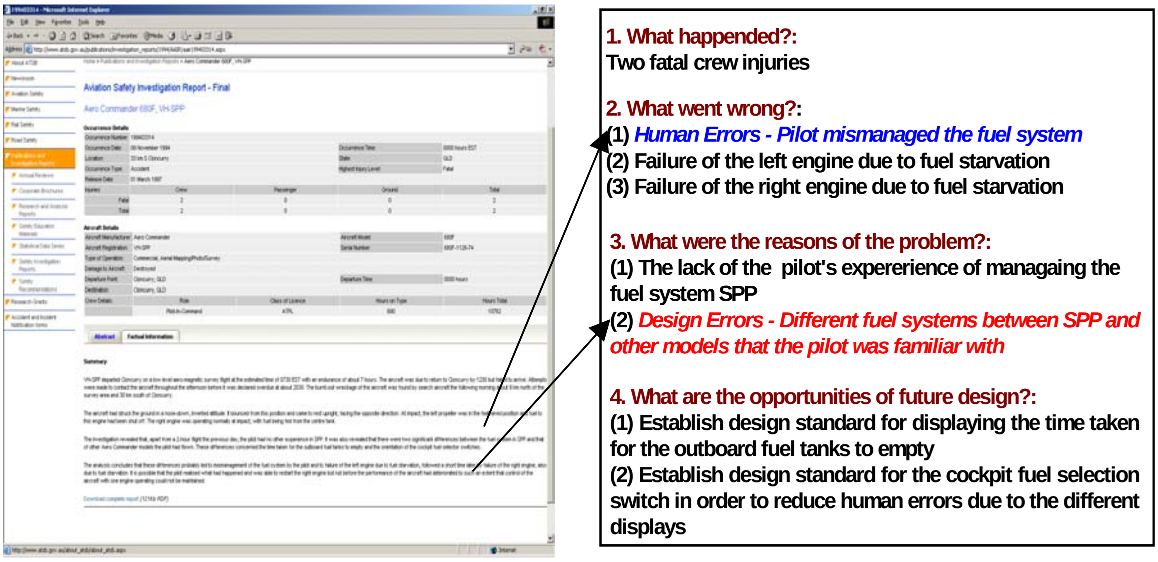
Cue phrase identification has been used when it is difficult to find such syntactic-lexical patterns [8, 9]. For example, by detecting the word *but*, a *contrast* discourse relation between two adjacent texts can be identified. Whereas it is easy to implement the cue phrase, due to syntactic and semantic variations, e.g. the word *mismanaged* can be rephrased as *did not properly manage*, relying entirely on cue phrases can lead to low coverage and ambiguity. To address this problem, Abdalla and Teufel proposed a bootstrapping approach that incrementally enriched each cue phrase with variants [8]. The cue phrases tested were pairs of transitive verbs and objects, e.g. *introduce* and *method*. While the method demonstrated high accuracy, it is not suitable for our task. Although the pairs of verb and object are useful, other types of cue phrases, i.e. nouns (*mismanagement*), or verbs without objects (*did not properly manage*), are needed in our case.

Text classification systems take a text as an input and assigns pre-defined categories to the text. To do this the whole content of the text is compared to the summary of each category. If the content is similar to the summary, the text is classified as belonging to that category. Recently, more studies have been conducted on sentence-level classifications. Sentence classification is the automatic classification of sentences into pre-defined sentence types. Example applications of such classifications are automatic text summarization and semantic orientations. The objective of summarization systems is to create a shorter version of an original text in order to reduce the time needed to read and comprehend it. The extraction of important sentences is one of the common tasks for summarization systems and there are two commonly used methods. The first method is to identify significance of words in the original text and to select a set of sentences based on the occurrence of high-scoring words [10]. The second method is to use adverbs and adjectives, e.g. *significant* and *hardly*, and to exploit the positions of the sentences in the text [11]. The performance of the automatic systems is often measured by comparing the automatic summaries with the summaries generated by humans.

In order to extract opinions, feelings, and attitudes expressed in a text, semantic orientation looks for the evaluative character of a word [12]. The orientation is classified as positive if it contains praise or recommendation. Negative orientation indicates criticism or non-recommendation. The semantic orientation does not apply to sentences that contain only facts. Wiebe and Riloff [13] proposed a classification method that classifies a sentence as subjective if the sentence expresses a positive or negative opinion, otherwise as objective. A combination of cue phrases, e.g. *excellent* or *low fees*, and linguistic features is commonly used. Those cue phrases can be created either manually or using a learning technique, e.g. PointWise Mutual Information (PMI)-IR or naive Bayes. On average, the accuracy is observed to be around 70%.

3 THE PROPOSED METHOD AND EVALUATION

Figure 1 shows how the proposed approach can help access the information in an accident report, especially establishing the causes of accidents and gaining insights. For example, two design opportunities are identified that could prevent the reoccurrence of the same Human Error. These are the establishment of two new design standards for: (1) the gauge displaying the time taken to empty a fuel tank; and (2) the switch for cockpit fuel selection. The example text used in both diagrams is an excerpt from the report of an accident that occurred in 1994. The text can be downloaded from from the Australian Transport Safety Bureau (ATSB) database http://www.atsb.gov.au/publications/investigation_reports/1994/AAIR/pdf/aair199403314_001.pdf.


Aviation Safety Investigation Report - Final
Aero Commander 680F, VH-SPP

1. What happended?:
Two fatal crew injuries

2. What went wrong?:
(1) *Human Errors - Pilot mismanaged the fuel system*
(2) Failure of the left engine due to fuel starvation
(3) Failure of the right engine due to fuel starvation

3. What were the reasons of the problem?:
(1) The lack of the pilot's expererience of managing the fuel system SPP
(2) *Design Errors - Different fuel systems between SPP and other models that the pilot was familiar with*

4. What are the opportunities of future design?:
(1) Establish design standard for displaying the time taken for the outboard fuel tanks to empty
(2) Establish design standard for the cockpit fuel selection switch in order to reduce human errors due to the different displays

Figure 1. An example of improved access to accident reports

3.1 Dataset

An evaluation of the proposed approach is based on the dataset provided by Shin et al [14]. The dataset contains 50 aviation accident reports downloaded from the ATSB database. The accidents occurred between 1994 and 2000. Each report was manually annotated and stored as XML format. A total of 3995 sentences were extracted from the 50 reports, and 208 sentences, i.e. 5% of the 3995 sentences, were tagged as either Human Error or Design Error. Of these, 108 sentences are classified as Human Error and 100 sentences as Design Error. Sentences that do not have such taggings are deemed irrelevant.

In the dataset, the sentences describing Human Errors or Design Errors were further classified using the sub-categories shown in Table 1. Human Errors have four categories, i.e. Decision Problem, Skill-level Problem, Distracted Cognition and Reliance on Systems. Table 1 also shows the number of times that each error was observed in the 50 reports. Such refinement provides a better means of determining analytically common human factors across accidents and supports a data-driven investigation. It also enables the identification of which types of Human Error are mostly triggered by Design Errors. This research is a part of an ongoing project [6]. As a starting point, the previous research developed a binary classification, i.e. a sentence is tagged either into Human Error or Design Error. The research has been extended to classify a sentence into multiple classes, i.e. a sentence is classified using one of three or more classes.

Table 1. The details of the dataset

Human Error	Number	Design Error	Number
Decision Problem	47	Modified Design	23
Skill-level Problem	43	Interface Design	72
Distracted Cognition	13	Work Environment Design	9
Reliance on Systems	6	Total	100
Total	108		

3.2 Text processing

The proposed approach takes accident reports as inputs, either by downloading them directly from aviation web sites or by accessing them from local copies. The reports are analysed and the sentences describing Human Errors or Design Errors are extracted. Engineering designers then use the analyses to search for the accidents caused by Human Errors and Design Errors. The proposed approach consists of three modules that: (1) pre-process accident reports and extract a set of sentences, (2) extract index terms from the sentences, and (3) classify the sentences into one of the pre-defined categories.

The first module pre-processes each document by dividing it into paragraphs each of which is decomposed into a set of sentences. Each sentence is analysed with shallow NLP techniques. NLP is known to improve indexing and retrieval accuracy compared to a string-based method. Since fully fledged NLP requires extensive linguistic resources and background knowledge, which are often difficult to acquire, shallow NLP techniques are commonly used. Terms are identified as words, numbers or combinations of them that are used to identify the contents of the document. Each sentence is first syntactically parsed using the Apple Pie Parser [15]. The Apple Pie parser generates a parse tree based on the grammars defined in the Penn Tree Bank corpus [16]. The parse includes the identification of POS taggings, e.g. *forgot* is tagged as *VBD* (past tense verb), and structure compositions of the sentence, e.g. noun phrases. POS taggings identify not what a word is, but how it is used. It is useful to extract the meanings of words since the same word can be used as a verb or a noun in a single sentence or in different sentences. In a traditional grammar, POS classifies a word into eight categories: verb, noun, adjective, adverb, conjunctive, pronoun, preposition and interjection.

The second module extracts index terms from the analyzed sentences and assigns a weighting to each of the index terms. Each POS tagged word is compared with WordNet definitions [17] to obtain term normalisation, i.e. stemming. Compared to a Boolean indexing that defines a feature only with its occurrence or absence, the weighting measures the quality of the indexing term in relation to an effective identifier in the sentence and specifies the numeric contribution. Term Frequency Inverse Document Frequency (TFIDF) is used for the weighting in this paper [18]. With TFIDF, the weights of terms in a sentence are assigned by their frequencies within the sentences multiplied by the inverse sentence frequency. This distinguishes the few sentences in which they occur from the many in which they are absent. In order to take into account different lengths of sentences, normalisations are performed.

The third module takes the annotated sentences as inputs and identifies patterns that distinguish sentence types from one another. The patterns constitute classification rules that are used for classifying new sentences.

3.3 Identification of using semi-fixed cue phrases

Cue phrases, either a single word, e.g. *forgot*, or multiple words, e.g. *high workload*, carry the semantics that are used to identify the types of sentence. Table 2 shows example sentences that were classified as Design Errors using the cue phrase *high workload*. Sentences in the first column indeed describe accidents caused by Design Errors whereas the sentences in the second column are falsely identified although their structures are quite similar to the correct sentences. The first column also shows a wide range of syntactic and semantic variations of the expression *high workload.*

Table 2: Correctly and incorrectly matched Design Errors sentences using high workload

Correctly matched sentences	**Incorrectly matched sentences**
The controller considered that the *workload was high* due to poor quality HF radio, increased coordination with other centres in relation to aircraft using 'flexible routes'	When an individual controller combines a number of positions, diverse scenarios and *increasing workloads* can quickly distract controllers
The *additional workload* created by instructions from ATC, and from attempting to re-program the GPS at the time when he was completing his climb checks may have captured his attention, thereby reducing his capacity to notice deviations from normal procedure.	The controller did not believe the *extra workload* generated by those tasks contributed to the occurrence.
In this occurrence, this led the aerodrome	AusSAR staff lacked familiarity with the

controller, at a time of *high workload* and possible stress, to forget that a conditional crossing clearance was pending.	capabilities and limitations of the Orion and the procedures for the in-flight re-tasking of aircraft by AusSAR were vulnerable to human error, especially under the very *high workload* occurring at the time of re-tasking.
The SURAD did not have identification labels or height information (facilities that were available on more modern equipment) and that limitation *increased the workload* on the controller.	He was responsible for two sectors of airspace but he did not believe that the *increase in workload* caused by the combination of the two sectors contributed to the error.

The dataset in Section 3.1 contains a list of potential phrases that can be used as cue phrases. Using simple regular expressions implemented in the Perl programming language, the list of phrases was analysed based on the results of the syntactic parse using the Apple Pie Parser. For some categories, a single word is sufficient for identification. The most commonly occurred pattern is the negated sentence, e.g. *did not properly manage* that is a syntactic pattern of <do or have> not <adverb> {verb}. The pattern has variations, e.g. <do or have> not {verb}, e.g. *did not manage*, <do, be or have> <adverb> not {verb}, e.g., *was apparently not managed*, or {verb} <adverb>, e.g. *managed badly*. Table 3 shows the list of the cue phrases. A total of 36 mutually exclusive cue phrases are pre-defined.

Table 3 Examples of cue phrases

Cue phrases for Human Errors		**Cue phrases for Design Errors**	
<do or have> not {assume, identify, interpret, notice, hear, detect, recognise} {misidentify, misinterpret, misunderstand, forget, omit}	Decision Problem	{modification, amended, update}	Modified Design
<do or have> not {update, conduct, scan, check} {inadvertently, incorrectly}	Skill-level Problem	<do or have> not {display, include, consider, alert, warn, provide} {absence, difficult, inadequate, ambiguous, deficient}	Interface Design
{distract, divert}	Distracted Cognition	{high workload, increased}	Work Environme nt Design
rely	Reliance on Systems		

Table 4 shows the classification results. Accuracy is used as an evaluation metric. It is calculated by dividing the number of correctly predicted classifications with the total number of classifications in the dataset. For example, 12 sentences were correctly identified as Decision Problem category that has 47 cases in the dataset, showing 26% accuracy. On average, 48% accuracy was observed for the four categories in Human Errors, and 35% accuracy for the three categories in Design Errors. The lowest precision was observed for the Interface Design category in Design Errors, implying that some of the cue phrases are not 'accurate'. For example, although the phrase *did not provide* is a good indicator to denote a problem caused by insufficient or ambiguous information, since it is also used to express potential effects of how related Human Errors can be generated (see Table 2), classifying a sentence by looking up the existence of this phrase can be misleading. A problem with such pre-defined cue phrases is that it is difficult to identify conditions under which the cue phrases are applicable or not applicable. For example, the sentences classified into Design Errors should implicitly or explicitly mention the designs or functions of the artefacts. Such constraints are particularly useful for classifying multiclass identifications. A supervised learning approach that learns classifications rules from tagged examples and is efficient at encoding various features is required.

Table 4: Accuracy results of error classifications using cue phrases

Human Error	Accuracy (%)	Design Error	Accuracy (%)
Decision Problem	26%	Modified Design	35%
Skill-level Problem	48%	Interface Design	19%
Distracted Cognition	69%	Work Environment Design	50%
Reliance on Systems	50%		

3.4 Multiclass classifications with Support Vector Machines (SVM)

Machine learning studies how computers can learn from observations, e.g. tagged examples, for performing intelligent tasks, e.g. text classifications. One type of learning is a supervised one that involves three tasks: (1) annotation of training examples; (2) classification rules that explain the training examples; and (3) generation of the classification rules to accommodate new examples. As a statistical learning theory, SVM is known to outperform other techniques, e.g. neural networks, due to its generalisation performance and its ability to handle high dimensional data. For a binary separation, examples are represented as *positive* or *negative*. The binary classification is extended into multi-class classification with one class versus all others approach, i.e. one class is positive and the remaining classes are negative. SVM attempts to find an optimal hyperplane that maximally separates the training examples into pre-defined categories. That is, it splits the positive examples from the negative ones by choosing the largest distance from the hyperplane to the nearest of the positive and negative examples.

We use the SVM^{multiclass} that is an implementation of the SVM in C programming language for multiclass classification [19]. It learns to predict one of *k* mutually exclusive classes. SVM^{multiclass} requires an example to be represented as pairs of features and their numeric weightings: <target> <feature>:<value> <feature>:<value>, etc. For example, *the pilot mismanaged the fuel system causing engine failure and fuel starvation,* is converted into *+7 pilot:0.1 mismanage:0.43 fuel system:0.12 cause:0.31 engine failure:0.023 fuel starvation:0.1.*

It is important to select only useful features when representing examples since some features might result in a noisy classification. Most of the previous work on feature selection uses term-specific feature representation. Recent studies have emphasized the importance of encoding the example sentences using various linguistic features [20, 21]. The proposed approach makes use of the linguistic features including: unigram (one keyword); bigrams (two keywords); POS taggings; the information on sentence constituents including subjects, verbs, and objects; active or passive sentence constructions; Named-Entities (NE); and verb tense. For example, the following features are encoded for the sentence *It was also revealed that there were two significant differences between the fuel system in SPP and that of other Aero Commander models the pilot had flown*:

Unigram: reveal, there, two, significant, difference, between, fuel, system, SPP other, Aero, Commander, model, pilot, flow
Bigrams: significant difference, fuel system, Aero Commander, Commander model, pilot flow
POS taggings: It/PRP, was/VBD, also/RB, revealed/VBN, that/IN, there/RB, were/VBD, two/CD, significant/JJ, differences/NNS, between/IN, the/DT, fuel/NN system/NN, in/IN, SPP/NNPX, and/CC, that/DT, of/IN, other/JJ, Aero/NNPX, Commander/NNP, models/NNS, the/DT, pilot/NN, had/VBD, flown/VBN
Subject: It
Verb: was revealed
Active/Passive: Passive
Verb tense: past

Subject: there
Verb: were
NEs: fuel system → Cockpit_Control_System
SPP, Aero Commander → Aircraft model
Pilot → Operator
Object: two significant differences…had flown.
Active/Passive: active
Verb tense: past

Both unigram and bigrams are extracted after removing common keywords, e.g. *it*, *or* and *also*, based on the pre-defined stoplists. Each POS-tagged keyword, e.g. revealed is tagged as VBD (past tense verb), is compared with WordNet definitions to achieve term normalization, e.g. revealed → reveal. Both keywords need special attention. A total of 31 725 unigram, including 23 639 bigrams, was extracted using the dataset in the Section 3.1. Not only it is difficult to make use of such a large number of the keywords, but also due to the noise in them, the identification performance can decrease if all the keywords are used for indexing. It is well known that feature selection improves the accuracy of a classifier. The feature selection deletes noisy features and reduces the feature-space dimension. In general, the first *n* features based on one of the ranking criteria are selected and are assumed to be more promising features for improving the identification performance. In this paper, the Information Gain (IG) measure is used for the feature selection. The top 1000 unigram and bigrams based on the IG values are used for the testing. NEs are the domain entities and the identification of NEs is a part of the IE tasks. For example, *SPP* and *Aero Commander* are the types of Aircraft Models. The identification of NEs is included in the dataset.

This experiment tested the performance of the classification rules generated by SVM. The dataset in Section 3.1 was equally divided into training and testing examples, i.e. 23 examples in Decision Problem category were used for training the SVM model and the remaining 24 examples were used for testing the model. Each sentence was represented using the features above. Table 5 shows the results.

Table 5: Accuracy results of error classifications by SVM

Human Error	Accuracy (%)	Design Error	Accuracy (%)
Decision Problem	35%	Modified Design	64%
Skill-level Problem	52%	Interface Design	72%
Distracted Cognition	67%	Work Environment Design	100%
Reliance on Systems	67%		

On average, the SVM achieved 55% accuracy on Human Error and 79% accuracy on Design Error. Compared to the cue phrases, the SVM showed a significantly improved performance, i.e. from 42% to 67%. In particular, the accuracy of identifying Design Errors increased from 35% to 79%. The lowest precision was observed for Decision Problem category in Human Error. The examination of the SVM model revealed that the classifications between Decision Problem and Skill-level Problem were quite confusing, i.e. the classification rules were not able to differentiate one from another. In a separate test, both categories were merged and tested with the same dataset. A significantly improved performance was observed, i.e. from 44% to 73% accuracy. No incorrect predictions were made in the Work Environment Design category.

The accuracy results in Table 5 demonstrate the potential of the proposed approach. Up to 79% of sentences expressing Design Errors were correctly identified and classified into three sub-categories. Using this approach, the accident analysts and designers can easily recognise recurrent and critical problems on the underlying designs and can develop better solutions to prevent the accidents. The approach also helps a data-driven analysis in particular for identifying the detailed types of human and design factor involved across accidents. Table 6 shows one example of such analysis. This analysis shows the number of co-occurrence between sub-categories in Human Error and Design Error. That is, it counts the number occurrences of aviation accidents which were contributed by one particular sub-category of Human Error that were caused by one sub-category of Design Error. For example, a total of 11 accidents were caused by Decision Problem that was influenced by errors in Modified Design. Among the Decision Problem, the most contributing design factor was identified as Interface Design, i.e. 24 occurrences of accidents. In particular, no accidents were made which have both factors of Reliance on Systems and Work Environment Design. This analysis demonstrates that the proposed approach helps establish better any links to Design Errors that contribute to Human Errors leading to improved understanding of the accidents. Such analysis is one of the areas where current other retrieval methods to the aviation accident reports are not suitable for.

Table 6: Analysis result on the number of accident occurrences using correlations between Human Error and Design Error

Human Error	Design Error	Occurrence of Accident/Incident
Decision Problem	Modified Design	11
	Interface Design	**24**
	Work Environment Design	6
Skill-level Problem	Modified Design	9
	Interface Design	24
	Work Environment Design	8
Distracted Cognition	Modified Design	3
	Interface Design	6
	Work Environment Design	5
Reliance on Systems	Modified Design	1
	Interface Design	4

4 CONCLUSIONS AND FUTURE WORK

This research has made progress towards the automatic acquisition of information from aviation accident reports. It has developed an approach for automatically classifying the information into one of seven categories of Human Errors and Design Errors. The approach not only helps engineering designers identify the impact of their designs on human operators, but also supports a data-driven analysis of which human factors are recurrent and critical. The use of linguistically enriched features for representing the sentences proved useful and showed an improved accuracy compared to the identification using cue phrases. Since the linguistic features were extracted using shallow NLP techniques, the proposed approach requires only limited linguistic experts' intervention. The contribution of this research has been to provide a level of detail necessary for practical applications. This has been achieved by moving from a binary classification to multiclass classification using SVMs. This research also shows that the proposed approach helps establish better any links to Design Errors that contribute to Human Errors leading to improved understanding of the accidents and gaining better solutions to prevent the reoccurrence of the same Human Error.

An extended evaluation of the proposed approach using a large number of accidents reports is planned. This evaluation will help test whether or not the approach is scalable and reliable. As a supervised learning method, the approach requires the reports to be annotated. In practice, it is expensive and labour-intensive to obtain annotated documents whereas it is easy to collect unannotated documents. An active learning approach that learns classification rules from unannotated documents is therefore being investigated.

ACKNOWLEDGEMENTS

This work was funded by the University Technology Partnership for Design, which is a collaboration between Rolls-Royce, BAE SYSTEMS and the Universities of Cambridge, Sheffield and Southampton. We specially thank InJae Shin of the University of Bath for the dataset and comments.

REFERENCES

[1] Berninger D. J. Understanding the Role of Human Error in Aircraft Accidents. In *Transportation Research Record No*. 1298, 1990.

[2] Bruseberg A. and Johnson P. Understanding Human Error in Context: Approaches to Support Interaction Design Using Air Accident Reports, In *12th Int. Symposium on Aviation Psychology*, U.S.A. 2003, pp.166-171.

[3] Johnson C. A. First Step Towards the Integration of Accident Reports and Constructive Design Documents. In *SAFECOMP*, 1999.

[4] Wiegmann D. A. and Shappell. S. A. A Human Error Analysis of Commercial Aviation Accidents Using the Human Factors Analysis and Classification System (HFACS), *Technical report of U.S. Department of Transportation*

[5] Shin I. J., Busby J. S., Hibberd R. E., and McMahon C. A. A Theory-based Ontology of design induced error. In *International Conference on Engineering Design, ICED 05*, 2003, Melbourne, August 15-18.

[6] Kim S. From design errors to design opportunities using a machine learning approach. In *6th International Conference on Practical Aspects of Knowledge Management, Austria*, 2006

[7] Grishman R. Information Extraction: Techniques and Challenges. In *Lecture Notes in Artificial Intellgence*, Vol. 1299., 1997.

[8] Abdalla R. M. and Teufel S. A Bootstrapping Approach to Unsupervised Detection of Cue Phrase Variants. In *the Association for Computational Linguistics (ACL)*, Australia, 2006.

[9] Knott A. and Dale R. Using linguistic phenomena to motivate a set of coherence relations. *Discourse Processes*, 1995, 18(1), 35-62.

[10] Strzalkowski T., Wang J. and Wise B. A Robust Practical Text Summarisation System. In *AAAI Intelligent Text Summarisation Workshop*, 1998. U.S.A., pp.26-30.

[11] Hovy E. and Lin C. Y. Automated Text Summarisation in Summarist. In *Advances in Automated Text Summarisation*, Mani, I., Maybury, M. (editors). 1999.

[12] Turney P. D. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *40th Association of the Computational Linguistics* (ACL), 2002, U.S.A. pp.417-424.

[13] Wiebe J. and Riloff E. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *International Conference on Computational Linguistics and Intelligent Text Processing*, (2005), Springer LNCS Vol. 3406 Springer-Verlag.

[14] Shin I., Kim S., Busby J. S., Hibberd R. E., and McMahon C. A. An application of semantic annotations to design error. In *IEEE International Conference on Hybrid Information Technology*, 2006, Korea, (in press).

[15] Sekine S. and Grishman R. A Corpus-Based Probabilistic Grammar with only two Non-Terminals In *Fourth International Workshop on Parsing Technologies,* 1995. pp.216-223.

[16] Marcus M. P. B., Santorini M. and Marcinkiewicz, A. M. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics,* 1994, 19(2). 313-330.

[17] Miller G. A., Beckwith R.W., Fellbaum C., Gross D. and Miller K. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 1993, 3(4), 235-312.

[18] Salton G. *Advanced Information-Retrieval Models*, *Automatic Text Processing, (Salton, G. Ed.)*, 1989, Addison-Wesley Publishing Company.

[19] Joachims T. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), 1999.

[20] Forman G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Technical Report*, HPL-2002-147R1 2002.

[21] Grimaldii M., Cunningham P., and Koraram A. An Evaluation of Alternative Feature Selection Strategies and Ensemble Techniques for Classifying Music. *Technical Report, TCD-CS-2003-21,* University of Dublin 2003.

Contact: Dr. Sanghee Kim
University of Cambridge
Engineering Department, Engineering Design Centre
Trumpington street
Cambridge
U.K.
44-1223-760559
44-1223-332662

shk32@eng.cam.ac.uk

www-edc.eng.cam.ac.uk/people/shk32.html