Natural Language Processing for aviation safety reports: from classification to interactive analysis

Ludovic Tanguy, Nikola Tulechki, Assaf Urieli, Eric Hermann, Céline Raynal

Natural Language Processing for Aviation Safety Reports: from Classification to Interactive Analysis

Ludovic Tanguy[a,*], Nikola Tulechki[a,b], Assaf Urieli[a,b], Eric Hermann[b], Céline Raynal[b]

[a] *CLLE-ERSS: CNRS & University of Toulouse*
[b] *CFH / Safety Data*

Abstract

In this paper we describe the different NLP techniques designed and used in collaboration between the *CLLE-ERSS* research laboratory and the *CFH / Safety Data* company to manage and analyse aviation incident reports. These reports are written every time anything abnormal occurs during a civil air flight. Although most of them relate routine problems, they are a valuable source of information about possible sources of greater danger. These texts are written in plain language, show a wide range of linguistic variation (telegraphic style overcrowded by acronyms or standard prose) and exist in different languages, even for a single company/country (although our main focus is on English and French). In addition to their variety, their sheer quantity (e.g. 600/month for a large airline company) clearly requires the use of advanced NLP and text mining techniques in order to extract useful information from them. Although this context and objectives seem to indicate that standard NLP techniques can be applied in a straightforward manner, innovative techniques are required to handle the specifics of aviation report text and the complex classification systems. We present several tools that aim at a better access to this data (classification and information retrieval), and help aviation safety experts in their analyses (data/text mining and interactive analysis).

Some of these tools are currently in test or in use both at the national and international levels, by airline companies as well as by regulation authorities (DGAC[1], EASA[2], ICAO[3]).

Keywords: Safety reports, aviation, NLP, document classification, text mining

*Corresponding author

Email addresses: tanguy@univ-tlse2.fr (Ludovic Tanguy), tulechki@univ-tlse2.fr (Nikola Tulechki), urieli@safety-data.com (Assaf Urieli), hermann@safety-data.com (Eric Hermann), raynal@safety-data.com (Céline Raynal)

[1] Direction Générale de l'Aviation Civile
[2] European Aviation Safety Agency
[3] International Civil Aviation Organization

1. Introduction

Air transportation, like other safety-critical activities, has seen the design and deployment of a large variety of safety-management procedures. Many of these efforts rely on a steady stream of reports that relate any abnormal event at any phase of activity and at any level of gravity.

This data is extremely valuable for learning lessons from past incidents and accidents, and hence for identifying new threats to safety and providing means of avoiding them. As in any complex system, the origin of these threats can be technical, organisational, environmental or human, or (most of the time) a combination of the above.

Because of this, national and international regulation bodies, as well as transport companies, store a large collection of reports for analysis.

Manual analysis of these reports is complex and requires considerable resources. Each safety event contains, in addition to other information, a description of the facts written in natural language, and each event is assigned codes from predefined taxonomies. Complexity arises, on the one hand, from the need to categorize the reports (given the size of the taxonomy, the users' knowledge, etc.) and, on the other hand, from the need to analyze and understand the reports from a global point of view. Our goal is to develop tools to help categorize and analyze the data.

CFH / Safety Data has been working on different aspects of these report systems for more than 10 years, in collaboration with the *CLLE-ERSS* linguistics laboratory. This paper is a wide-spectre presentation of the joint research we have conducted in order to integrate natural language processing (NLP) tools in the management of aviation safety reports. This work has been performed in close collaboration with both the data providers and the end users (safety experts). This article is organised as follows:

Section 2 presents a synthetic view on the existing aviation safety reporting systems and data, and summarizes the different tasks that have been identified for NLP to fulfil.

In Section 3 we present the most straightforward of our approaches: the *classification* of reports. A classical problem for NLP, it can quite easily be dealt with using supervised machine learning techniques based on textual content, and we show it succeeds when non-extreme conditions are met. Although this method is currently used by some companies and authorities, this solution is limited by the classification process itself, which is not adapted to a constantly changing environment and cannot be used for the identification of emerging threats.

We propose to address this problem with inductive methods that aim to mine patterns in the text data, and lead to the proposal of categories that can be compared to existing ones. We describe in Section 4 an experiment with *probabilistic topic models* on a large collection of reports, with mixed results that cannot conclude on the utility of this method in the specific case of extensively described and annotated data such as the repositories used in aviation safety management.

In the following two sections we present how specifically designed *interactive* tools can be useful to assist experts in their exploration of these huge and complex databases.

In Section 5 we propose a method based on the notion of document content similarity. The *timePlot* search tool is already used by several safety experts in France and enables them to quickly identify reports that are similar to a target occurrence and thus to find possible antecedents to a single event.

The last approach (Section 6) uses an *active learning* procedure in order to assist an expert circumscribe a known but not thoroughly defined aspect of incidents. Contrary to a fully inductive statistical approach, it is based on the ability of an expert analyst to quickly define the raw contours of a target category of threat. We present a proof of concept of this method that encourages us to propose such a solution to safety analysts.

In the conclusion, we discuss the extension of these techniques and processes to other fields of activity, and adress the delicate problem of evaluating such techniques.

2. Overview of aviation safety reports

In 2012 the probability of dying on a single flight on one of the top 39 airlines was one in twenty million. Indeed, safety in air travel is constantly improving. ICAO (2013) reports 2012 as the year with the lowest accident rate (3.2 accidents per million departures) since they started keeping the record. In the vast majority of cases, even when something serious, such as an in-flight engine malfunction, occurs, the accident is avoided and the aircraft lands safely. Even more often something could have happened but was avoided in time thanks to specific equipment, training or safety procedures.

All of these reassuring facts are the results of constant efforts at improving safety at every level of the complex system that enables air transportation. One of the procedures that helps define appropratie safety measures is incident reporting.

2.1. Principles of incident reporting

Incident reporting is a large-scale process that enables (encourages, and sometimes requires) parties to relate any abnormal event (or occurrence) to a central entity that collates and then uses this data for safety prevention purposes. This is mostly done in a non-punitive manner, i.e. the purpose is not to blame the person making the report, even if he admits that he made a mistake at some point. Quite on the contrary, such feedback loops help the personnel feel directly involved in the safety process. It should be noted that in some case parties are also invited to share their positive experience given that this kind of feedback (adequate procedure, team work, etc.) is as important as problematic events to improve safety.

Johnson (2003) identifies several arguments for setting up such a procedure, the main ones being:

- incident reports indicate why an accident did not occur, and help identify both the sources of danger and the safeguards;
- incidents are much more frequent than accidents, and can be submitted to quantitative analyses, giving insights into the main sources of danger;
- the data obtained is cheap—much cheaper than the cost of accidents, especially in the industrial and transportation sectors.

In addition to these obvious advantages, regulatory decisions may compel civil aviation companies or administrations to set up a reporting system. Indeed, in most countries, reporting serious incidents to the regulation authority is mandatory.

The exact architecture of a report system varies from simple centralised repositories to complex control procedures and feedback loops, but the minimal structure is as follows:

1. The reporter writes a relatively free-form text describing the incident, along with a small set of metadata (mostly concerned with the time, the location and the equipment involved) and generally assigns a category (see below).
2. The report is checked by a receiver who assesses its compliance, and sometimes add comments, remarks and/or metadata.
3. The report is stored in a database where it is indexed according to its metadata.
4. Analysts access the database in several different ways, ranging from simple queries and statistics that estimate the frequency and evolution of incident types, to data-mining investigations in order to identify emerging dangerous situations.

As for any collection of data, organisation and indexing are crucial to its usefulness. However, the very nature of the reports' origin makes it difficult to correctly organise and index them. The spontaneity of their writing by anonymous personal (as anonymity is an important part of the non-punitive aspect of incident reporting) and the large number of reports are the two main obstacles that analysis procedures have to deal with.

2.2. Sample systems and data

Although many different reporting systems exist at different levels for companies, government agencies and NGOs, we present two of the most widely used. ASRS is a North-American database of incident reports, while ECCAIRS is a software system proposed by Europe for managing incident reports at different levels.

2.2.1. ASRS

ASRS (Aviation Safety Report System) is the oldest and most famous voluntary incident reporting program for aviation. It is managed by NASA and collects voluntarily-submitted reports of aviation events in the United States[4]. Operational since 1976, ASRS has processed over a million incident reports and averages 6736 submissions monthly (322 daily), with an increasing rate over the years. This system targets several types of events from different types of reporters: general reports from pilots, Air Traffic Control reports from controllers, maintenance reports from mechanics and cabin reports from cabin crew.

At the end of the intake process a typical ASRS report consists of one (or several) narrative fields and a set of descriptors. The narrative fields correspond to the report submitted by the author(s) and a summary (or synopsis). The first set of descriptors that accompany the textual part of the report provide detailed objective information about the location, time, weather conditions, equipment and people involved, etc. In addition, more interpretative descriptors are used to described the event with controlled values (categories); these and the synopsis are coded by the ASRS experts upon reception and analysis of the occurrence.

A sample narrative is reproduced in Figure 1.

> As we made our approach and landing into RAP we were cleared to exit Runway 32 via Runway 5 and taxi in via Alpha. Once we landed in the rain the visibility quickly diminished while trying to exit the runway. We inadvertently turned onto Taxiway B which is approximately 40-60 feet prior to the Runway 5 intersection. Taxiway Bravo is supposed to be closed due to new concrete that was poured but the taxiway was only barricaded from taxiway Alpha and not the Runway 32 side. So we ended up halfway on Bravo. The lack of proper markings and barricades combined with the late evening hours and poor weather conditions led to our wrong turn. Luckily we were able to prevent any further incident and get tugged back on the runway and taxied on the specified taxiways.

Figure 1: Sample ASRS report narrative (ACN 1189955)

The descriptors follow a strict list of values that are hierarchically organised in a two-level taxonomy and are of paramount importance when analyses are performed. Aspects of the incident are grouped around *entities*. Each entity represent a logical grouping of descriptors. Aspects dealing with the aircraft, for example (make/model, operator, flight phase, filed flight plan, etc.) are grouped into the *Aircraft* entity. A separate *Person* entity is created for each person that played a role in the incident. In the report in Figure 1, for example, two people were involved (the captain and the first officer) and for each one, information such as their experience, function and location in the aircraft is coded. Also, since 2009, human factors information relating to each person is also coded at

[4] `http://asrs.arc.nasa.gov/`

this level. In the above mentioned example, the captain was *confused* by the markings and this information is coded in an attribute to the relative *Person* entity. The *Events* entity concerns information about the events that took place. In the example in Figure 1 eight attributes in the *Events* code that there was a *ground incursion* during *taxiing*, that it was detected in time by the flight crew and that, as a result, they requested clarification from the tower and became reoriented. Finally an *Assessments* entity summarises the incident, stating the primary problem and the identified causes and contributing factors. The above example is analysed as an *ambiguous* incident where *weather*, *human factors* and *issues at the airport* were contributing factors.

Given that ASRS capture data since the 1970s, the form of the reports evolved considerably over time. In roughly the first two decades of its existence, the system imposed a particular writing style to the report narratives. Rather than writing in standard English, the reports were keyed in using a semi controlled and standardised language, making heavy use of abbreviations for common aviation terms such as *ACFT* for aircraft and *WX* for weather. The reports were also written using only capital letters. Figure 2 shows an example of this writing style, along with its "translation".

FLT (flight) WAS SBND (southbound) ON J-209 AND HAD BEEN CLRED (cleared) TO FL390[a] BY A PREVIOUS CTLR (controller). OVER SBY[b] VOR[c], CLIMBING THRU (through) FL360, TFC (traffic) WAS CALLED BY ZDC[d] NBOUND (northbound) AT FL370 AND 4 MI (military) TFC (traffic) WAS OBSERVED AND CENTER THEN HAD US DSND (descend) TO FL350.

[a] **F**light **L**evel **390**00 feet
[b] Salisbury–Ocean City–Wicomico Regional Airport
[c] VHF Omni Directional Radio Range navigation system
[d] Washington Air Route Traffic Control Center

Figure 2: Sample ASRS report narrative using the old writing style (ACN 145677)

Variations in style is inherent to the incident reports systems, and their is a wide continuum between the two sample reports reproduced here, across both time, culture and authors. Part of the task of managing this data is to cope with such disparity.

ASRS data is public and can be queried like any traditional database, through an online form in which the user expresses boolean restrictions on the descriptors, in combination with a simple word search in the narrative parts[5]. ASRS is the world-wide reference for incident reporting systems, frequently cited as an example even for other sectors of activity such as medicine (Helmreich, 2000).

[5] Altough the proposed text search utility is rudimentary, and does not even provide a correspondence between "CTLR" and "controller", which have both to be entered in the engine in order to achieve reasonable results.

2.2.2. ECCAIRS

ECCAIRS (European Coordination Centre for Accident and Incident Reporting Systems) is an ongoing effort at standardising accident and incident data collection and exchange within the European Union (Menzel, 2004). Developed by the European Commission's Joint Research Center, ECCAIRS's mission is "to assist national and European transport entities in collecting, sharing and analysing their safety information in order to improve public transport safety" and is freely available to any interested party. It takes the form of a software platform that covers most of the collection, indexing and querying of incident reports. It is currently used by several national agencies, among them the French DGAC[6].

The reports collected by DGAC are similar to those of ASRS, with the notable distinction of being written in several languages (French and English). A sample report is shown in Figure 3, with specific data (location, company, makes, etc.) anonymised as requested because the data is not publicly available.

> Weather was forecasted with snow and moderate icing in [*LOCATION*] area. Runway 09R was closed in [*AIRPORT1*] according to ATIS broadcast. During final 09L, preceding aircraft advised TWR of slippery runway, what led TWR to ask us to go around for a standard procedure to [*AIRPORT1*]. Several holding patterns were turned over [*AIRPORT1*]. Some time later, on our request, approach advised us of an expected 10 minutes delay. At that point, fuel on board was around 1180 kg including divertion reserve of 970 kg (in icing conditions) to [*AIRPORT2*], where the runway was declared only wet. Roughly, these 210 kg of extra fuel represented slightly a little more than 3 turns over [*AIRPORT1*]. Holding 10 more minutes would have been led us very close to minimum fuel, as a result of which captain decided not to wait and divert to [*AIRPORT2*]. Fuel trip to [*AIRPORT2*] was much smaller than expected (200 kg) thanks to direct vectors. At the end of radar down wing leg to runway 09 [*AIRPORT1*], ATC advised that 09L in [*AIRPORT1*] was declared open again. Normal landing in [*AIRPORT1*] with 980 kg of fuel onboard instead of 630 kg calculated over [*AIRPORT2*]. ATC was professional and helpful.

Figure 3: Sample DGAC report in ECCAIRS

In ECCAIRS also, additional information is represented by controlled metadata attributes. The taxonomy followed is ICAO's ADREP[7] (ADREP, 2010).

The ADREP taxonomy is the result of an effort at standardisation of aviation incident and accident information supported by ICAO (Stephens et al., 2008) and is intended for a very broad coverage. Unlike ASRS's taxonomy, ADREP is before all an international standard and thus needs to potentially adapt to

[6]Direction générale de l'aviation civile
[7]Accident/incident Data REPorting

every possible situation and scenario. Similar to ASRS, both factual descriptors (time, place, aircraft models, engine and component manufacturers etc.) and information resulting from the expert's analysis of the occurrence, such as *event types* and *contributive factors* are organised in a complex multilevel hierarchy with more than 800 attributes and 160,000 possible values.

The ADREP taxonomy has proven to be very useful when used correctly, facilitating data exchange and providing a common frame of reference when speaking about incidents and accidents in aviation (Stephens et al., 2008). However most of the time, fine-grained categorisation is simply not available, as in the case of the DGAC database we are working with, where only a third of the occurences are coded with the *occurrence category*, and even less for more precise information such as *event types*, the main branch in ADREP for abstracting information about the precise sequence of sub-events that occurred.

The *occurrence category* attribute has a closed list of 37 values providing high-level classifications for occurrences. The occurrence in Figure 3, for example, is classified as an *ATM*[8] and a *FUEL* related occurrence. This main category is the target of the classification system described in Section 3.

Every European country maintains an ECCAIRS database, and these are merged at the community level by EASA[9]. The ECCAIRS software platform allows for complex querying of the databases, with a clear focus on helping the user manage the complexity of the taxonomy, at the expense of textual search. Unlike ASRS, ECCAIRS databases are not public and their target users are safety managers and analysts.

2.3. Identifying tasks for NLP

Having presented an outline of the data gathered and managed in incident report systems, we now take a closer look at their usage.

According to Johnson (2002):

> There are two central tasks that users wish to perform with large-scale incident reporting systems. [...] On the one hand, there is a managerial and regulatory need to produce statistics that provide an overview of how certain types of failures are reduced in response to their actions. On the other hand, there is a more general requirement to identify trends that should be addressed by those actions in the first place.

In other words, the user should be able to quantify any relevant aspect of an event (cause, effect, factor, etc.), study its evolution in time and space, and query the link between several factors. He should also be able to identify new configurations that have led to dangerous situations, and to detect the emergence of new problems.

[8] Occurrences involving Air Traffic Management (ATM) or communications, navigation, or surveillance (CNS) service issues.

[9] European Aviation Safety Agency

These tasks are quite straightforward, as long as they rely on techniques applied to the metadata. Indeed, it is easy to get an overview of the frequency of incidents implying ground markings if such an aspect is clearly encoded in the database. Analysing this aspect as a function of other characteristics (airport, time of the day, airplane model, etc.) is also non-problematic. Identifying correlations with another aspect of an incident (crew fatigue, etc.) can be done with standard data-mining techniques. This aspect of the task is not within the scope of our approach.

In any event, the feasibility of these tasks primarily relies on the taxonomy-controlled descriptors that summarise the reports' content and these important features need to be hand-coded, either by the reporter or by a safety manager. Given the flow of incoming reports and the complexity of the taxonomies, this can be a costly and difficult task, and classification errors can easily occur, with serious effects downstream

We thus identify a first task where NLP techniques are useful: since the essentials of the event are described in a natural textual form, it should be possible to infer some of these metadata descriptors from the narrative. This task of automatic text classification is presented in Section 3.

However, taxonomies are limited in the sense that categories are generally too broad to allow the identification of a specific characteristic of an event (Johnson, 2002). For example, an expert might be interested in improper marking and signalisation in an airport (as evoked in the sample report in Figure 1). In the ASRS taxonomy, the corresponding metadata feature is *contributing factor:airport*, and a query on this field would obviously lead to a large quantity of noise. In the more detailed ADREP, we can find the following hierarchy:

```
Event >
   Aerodrome & ground aids >
      Aerodrome systems >
         Markings >
            Apron marking
            Runway marking
            Taxiway marking
            Obstacle marking
```

In reality, the metadata rarely descends this deep into the hierarchy, staying at the upper two levels of the ADREP taxonomy as noted before. Besides, this level of detail is generally accompanied by a greater difficulty for the human coder to choose between closely related values (such as Taxiway and Runway markings in our example), and leads to unreliable metadata with a poor inter-annotator agreement (Johnson, 2002).

The alternative is to rely on the narrative part of the report, in which the explicit information is given in textual form. Our question is thus: is it possible to use the variety of expressions found in the narrative parts to identify stable categories? These categories can either match metadata, or be more fine-grained, or even help new previously ignored patterns emerge. This inductive building of categories can be done by statistical methods such as topic modeling, which we present in Section 4.

However successful these methods turn out to be, they still cannot systematically provide a solution to the specific needs of an expert, nor to the very fine-grained investigation of emerging sources of danger. In order to do this, we must place the expert at the heart of the process, and have him in control of the access to the database, along with his thorough knowledge of both data and domain.

Navigation in these huge databases is a complex task, and although full-text retrieval can be useful, its results are not always satisfactory on such heterogeneous data. Based on the experts' expression of their needs, we have designed a similarity-based tool that enables the user to visualise and access the reports similar to a single occurrence selected by the user. This approach and the *timePlot* tool are described and discussed in Section 5.

Moreover, browsing the database and identifying occurrences cannot lead to an operational answer to the tasks evoked by Johnson. Certain aspects of the incidents such as phenomena relating to human factors (fatigue, confusion, stress, etc.), are particularly elusive with respect to full-text searches, and require a more global approach than the one provided by incident similarity. These aspects need to be clearly identified and automatically marked, thus enabling the automation of their retrieval and their inclusion in statistics and data-mining investigation techniques. We propose to use a semi-automated technique based on active learning that enables the user to iteratively design a classification model that efficiently isolates the target aspect. We describe this technique and exemplify its uses in Section 6.

The tools and experiments presented in this article cover only certain aspects of how natural language processing techniques could be used in the domain of safety incident reports: indeed, Andréani et al. (2013) have identified that NLP techniques can be useful at any of the phases of an incident report lifecycle, from the initial reporting to the analysis.

3. Automatic report classification

In this section we present automatic document classification techniques to improve the usability of large databases of incident reports. The principle of classification is to assign a category from a closed list of possible values (here a taxonomy-constrained metadata) to an item (here a report) according to its characteristics (here its textual content). This classical task is usually accomplished through the use of a supervised machine learning algorithm that constructs a model of the task by observing a volume of previously categorised data.

Such a tool was developed by *CFH / Safety Data* and used on a French airline company's internal occurrence reporting database as described in Pimm et al. (2012). It consists of a system based on learning the correlations between automatically extracted linguistic descriptors and coded values from the airline's

SMS[10] taxonomy. Each time a new occurrence arrives, the system calculates one or more categories to be assigned and proposes them to the safety expert in charge of indexing the report. Although not perfect, this system can easily identify the most common situations and thus preserve the expert's available time for more uncommon and potentially more dangerous occurrences.

Here we present a similar system designed to be used on the French DGAC's database of incident and accident reports. We describe the learning mechanisms and evaluate different configurations on real data.

3.1. Context

The DGAC is France's national aviation regulator and collects occurrence data from a variety of entities operating on French territory. Mandatory reporting policies dictate that aviation incidents must be recorded by companies, airports and air traffic controllers and forwarded to the DGAC's central database. The implementation of reporting policies in France is very successful today, in terms of number of reported incidents and accidents. This makes France the most productive contributor to the EASA european database (DGAC, 2013). The database we are working with contains more than 400,000 occurrences collected over the past ten years, with approximately 45,000 incoming reports per year, a number constantly on the rise. Reports are mostly written in French (97%), although their authors make heavy use of technical aviation terms borrowed from English.

The DGAC uses the ECCAIRS environment and the ADREP taxonomy for managing their occurrence data. As we already pointed out in 2.2, ADREP provides a very detailed scheme for incident categorisation, using an elaborate hierarchy of descriptors. The task of categorisation is highly time-consuming and, given the volume of reports, is very demanding for the safety experts.

One of the branches of the ADREP taxonomy is the *occurrence category*[11] providing a high-level description of the corresponding event. In theory, every event can be reliably categorised using one or more of the 37 labels. A consistently labelled database would allow safety experts to examine trends and statistics based on the labels, as well as filtering incident searches by label.

Like the rest of the ADREP taxonomy, the labels themselves are normalised and are associated with a set of conditions that describe when they should be used. Table 1 shows some of the categories with their associated descriptions and their relative frequency.

3.2. Corpus size and category distribution

The DGAC's database currently consists of 404,289 occurrence reports from 2004 until September 2014. Among these, only one third are labelled with

[10] Safety Management System.

[11] A slightly out-of date list of all the categories and the associated descriptions and usage notes is available at `http://www.skybrary.aero/index.php/Occurrence_Category_Taxonomy`. The latest version is available as part of the ECCAIRS software package at `http://eccairsportal.jrc.ec.europa.eu/`.

Label	Description	% reports
ATM	Occurrences involving Air Traffic Management or communications, navigation, or surveillance (CNS) service issues.	40.6
BIRD	Birdstrike - Occurrences involving collisions / near collisions with birds.	7.3
RE	Runway excursion - A veer off or overrun off the runway surface.	0.7
GTOW	Glider towing related events.	0.03

Table 1: Examples of ADREP occurrence categories

at least one occurrence category. We limited our study to reports written in French. The corpus used in our study thus contains 136,861 documents, which amount to a total of 15 million words.

The categories themselves are very unevenly distributed as can be seen in the examples in Table 1. The most common category is *ATM*, assigned to 40.6% of the corpus, while 25 of the 37 labels concern less than 1% of the reports. Some categories are very poorly represented: for example, *GTOW*, the category concerning glider-towing related incidents, concerns only 46 reports or 0.03% of the corpus (in addition to the rarity of these events, this category was recently added to the taxonomy).

The ADREP scheme considers that an occurrence can be described with more than one label, which leads to a multi-label classification situation. Among the labelled reports of the database, 95% have one category, 4% have two categories and only 1% have three or more (maximum 6).

3.3. Features and training

We used the Support Vector Machines, or SVM (Fan et al., 2008) supervised learning algorithm, as this technique has proven to get excellent results for document classification tasks (compared to alternatives we have also tried, such as Maximum Entropy). However before applying SVMs, text data has to be transformed into features describing the content of each report with numerical values.

After extracting the textual parts (narratives) of the 136,861 French categorised documents from the DGAC corpus we applied a custom rule-based normaliser developed by *CFH / Safety Data*, and based on the Talismane NLP toolkit[12] (Urieli, 2013). This normaliser currently comprises 637 hand-written rules that fold some frequent common variations to a standard term. In these reports, common terms such as "take-off" for example can have multiple variants ("T/O", "take-off", "takeoff", "T-O", "take off"). Other multi-word terms

[12] `http://redac.univ-tlse2.fr/applications/talismane`

such as "check list" and "glide slope" follow the same pattern of variation and are folded to a single term by the normaliser[13]. All numerals are also normalised to a generic token "#NUMBER#", all characters are folded to lower case and accentuated characters are replaced with the corresponding deaccentuated ones.

After this normalisation, we constructed the following text units:

- **words**: The words from the text, as detected by a standard white space and punctuation *tokenizer*.
- **stems**: Word stems as detected by the *Snowball stemmer*[14] designed for standard French.
- **character n-grams**. All substrings of n characters contained in the text. We limited n to 3 and 4.
- **stem n-grams**. All sequences of n contiguous stems contained in the text. We limited n to be between 2 and 6 and extract only those sequences that don't start or end with a function word such as a determiner or preposition.

The total number of features reaches 8 million in this experiment. The number of unique features for each type is indicated in Table 2 along with the total number of occurrences and the sparsity (average frequency per feature). Longer stem n-grams are limited in number due to the constraint on function words. No feature selection was performed based on frequency for this experiment.

Feature type	**Unique**	**Occurrences**	**Sparsity**
stem	156,176	14,637,737	93.73
word	182,931	14,637,737	80.02
characterNgram-3	132,664	82,915,335	625.00
characterNgram-4	742,270	82,647,113	111.34
stemNgram-2	689,105	3,180,313	4.62
stemNgram-3	1,502,468	3,291,545	2.19
stemNgram-4	1,759,628	2,632,952	1.50
stemNgram-5	1,703,677	2,128,725	1.25
stemNgram-6	1,529,965	1,778,064	1.16

Table 2: Breakdown of features

Once these units are clearly identified, we computed the relative frequency of each unit in each text and thus got a representation of each report as a vector of numerical features. The sets of units described above can be combined and allow different feature configurations to be considered.

Training (i.e. the construction of the predictive model) was performed with the java port of the *Liblinear* library[15] (Ho & Lin, 2012). As this is a multi-label classification problem, we in fact trained 37 independent binary classifiers,

[13] As previously stated, reports written in French make heavy use of English terminology.
[14] `http://snowball.tartarus.org/`
[15] `http://liblinear.bwaldvogel.de/`

one for each target category. This means that each report to be categorised is analysed by these 37 classifiers, and given an independent yes/no answer for its association with the 37 possible categories.

We used a linear kernel for these SVM classifiers. Using only the *words* set of features we performed a grid search on a single fold of a 10-fold experiment (see below), finding the optimal parameters C=0.5 and $\epsilon = 0.1$.

3.4. Evaluation

In order to evaluate the classifier we performed a ten-fold cross validation. From the 136,861 documents we constructed a training corpus consisting of 90% of the documents and kept the remaining 10% of the documents for testing. We repeated this partitioning 10 times so that each document is present exactly once in the test corpus.

Using the setup described above, for each run we trained several sets of classifiers using different combinations of features. For each combination we calculated the micro-average precision, recall and F1-score (i.e. considering the assignment of a category to a document as an individual event). Table 3 summarises the average results for several combinations over the ten runs.

Feature combination	**P (%)**	**R (%)**	**F1 (%)**
words	84.95	71.08	76.46
stems	84.61	73.10	77.92
words + stems	85.21	71.03	76.60
stems + sn3	**86.79**	**74.08**	**79.15**
stems + sn3 + cn4	85.74	72.12	77.55

Table 3: Evaluation of different feature combinations

The best performing combination uses *stems* and stem n-grams of length 2 and 3 (sn3). The overall results are encouraging with a F1-score of nearly 80%.

In terms of features, these results show that using a stemmer produces better results than using non-normalised words, and that trying to combine stemmed and unstemmed words makes the results even worse. Adding character n-grams (cn4) also worsens the performance of the classifier.

We had a closer look at the results of our best configuration. First of all, we found obvious inconsistencies in the original coding. One of the errors we identified was a common confusion between some of the categories and the *OTHR*[16] category. When looking through the errors concerning the *RAMP*[17] category we identified that events concerning spillage of fuel while refuelling were (correctly) classified by the tool as *RAMP* events, while in the training

[16]Other - the catch-it-all category defined as "*Any occurrence not covered under another category.*"

[17]Ground Handling - Occurrences during (or as a result of) ground handling operations.

corpus, roughly one out of five[18] such events had been attributed the *OTHR* category.

Another common classification error is between the *LOC-G* (Loss of control on the ground), *ARC* (Abnormal runway contact) and *RE* (Runway excursion) categories. All three concern events that happen upon touchdown, and as such share a number of expressions (*runway*, *touchdown*, *landing gear*, etc.). *LOC-G* is meant to be used if the crew actions leading to the loss of control were posterior to the moment of touchdown. *ARC* is used for event where the landing was abnormal. In both cases, if the aircraft at some point left the runway, *RE* is the correct code. When looking closely at the classification mismatches between these three categories, missing or incorrect codes in the evaluation corpus are as common as automatic classification errors.

Category	**Count**	**P (%)**	**R (%)**	**F1 (%)**
ATM	55614	96.31	93.09	94.67
BIRD	9943	96.08	93.01	94.51
MAC[19]	7503	91.54	84.72	87.99
SCF-NP[20]	9529	80.31	62.42	70.18
SCF-PP[21]	2530	72.15	53.92	61.68
RE	943	87.62	77.61	82.04
GCOL[22]	850	59.62	36.47	45.26

Table 4: Detailed scores per category

Table 4 shows detailed results of the classifier's performance for various categories. It appears that our classifier gets very good results (with a precision exceeding 90%) for several categories, among which we can find some that are very frequent, such as *ATM* and *BIRD*.

Other categories are inherently difficult, even when frequently used. There are many components in an aircraft and they all may fail. The (non-powerplant) system component failure category *SCF-NP*, whose frequency is comparable to the bird strike category, is much more difficult to recognise. The difficulty comes partly from the fact that a component failure will constitute a larger event and the crew's actions (such as declaring an emergency, troubleshooting the error jointly with ATC) will be reported. This surplus of information creates a much harder problem to solve for the classifier.

Finally, while data rarity is an obvious issue when considering machine learning approaches, it has not been too problematic in the present study. The *RE* category, for example, concerns only 94.3 occurrences on average and is classified with relative reliability. For other rare categories, such as *GCOL* (ground

[18] Determined by a manual examination of 200 documents.
[19] MAC: Airprox/ACAS alert, loss of separation, (near) midair collisions.
[20] SCF-NP: System/component failure or malfunction [non-powerplant].
[21] SCF-PP: powerplant failure or malfunction.
[22] Ground Collision.

collision) the performance is much worse and can be attributed to a combination of rarity, difficulty[23] and inconsistency[24].

3.5. Usage scenario and limits

Based on these results, we can extend the usage scenario already in use for the airline's database to the ADREP metadata scheme. More precisely, the categories for which our classifier reaches the 90% efficiency threshold can be proposed in a computer-assisted coding of the incoming reports. Those for which precision exceeds 95% are even considered to be processed without human verification. This means that some events are sufficiently stable through their corresponding reports that we can free the experts from addressing them through a complete reading of the report. This allows them to focus on more specific cases that cannot be satisfactorily managed by automated means. There is no absolute threshold for assessing the reliability of such a system, but it should be mentioned that some training schemes only require the trainees to reach 75% accuracy (Johnson, 2003, p. 768).

Although we did not report the experiments here, similarly good results have been achieved for other metadata such as flight phase (cruise, landing, etc.) and occurrence class (accident vs incident). This confirms that simple NLP techniques such as the ones used here can be easily applied to this situation.

On the other hand, the other parts of the ADREP coding scheme are out of reach of such techniques. The next descriptive branch, *Event types* has more than 1,600 categories (dispatched on 3 hierarchical levels). Data sparseness is of course the major obstacle here, as well as a lower quality of the data available at this level, given how complex it is even for an expert to clearly and unambiguously identify the exact tag to be used. Such low-quality unbalanced training data generally means that machine learning is a waste of time, especially given the quality requirements of the safety management process.

In this section we presented how machine learning can be used to classify documents according to predefined categories. We also hinted on how taxonomies in general, and ADREP in particular can be misused and altogether ill-suited for the particular safety-related task at hand. Nevertheless taxonomies are essential as they provide the necessary abstraction for data-driven safety management. Unlike the the aviation sector, most other sectors lack normalisation efforts such as the one producing and maintaining ADREP. Meanwhile, textual descriptions of safety-related events are piling up by the thousands. In the next section we will present how *probabilistic topic modelling* addresses the data abstraction problem in a bottom-up manner by determining the thematic structure of a

[23] There are several categories dealing with collisions.

[24] When reviewing the data, we are convinced that this particular category is largely under-represented: there are many events that should be coded *GCOL* and are not.

(large) corpus of texts. In this sense topic modelling has the potential to serve as a first step when one is designing a taxonomy for a particular sector.

4. Topic modelling of incident reports

Probabilistic topic modelling is a generic method initially designed by David Blei (Blei et al., 2003; Blei, 2012). Following older methods of documents representation such as Latent Semantic Indexing (Deerwester et al., 1990), its main purpose is to represent a collection of documents in a vector space with a reduced number of dimensions or *topics* (as opposed to traditional vector spaces where each dimension corresponds to a single term or word). These topics or latent dimensions are calculated without any kind of supervision or external knowledge, based solely on the distribution of words in the documents. Thus, the topics are supposed to be a good representation to the underlying thematic structure of the collection.

Topic modelling has attracted considerable attention from the NLP community in the past decade, and has been used in a number of applications ranging from information retrieval and document classification to the summarisation of the main themes addressed in a document collection. It is this specific use that led us to applying them to incident reports. Former successful experiments have been run on collection of scientific publications (Hall et al., 2008), newspaper articles (Newman et al., 2006) and encyclopedia entries (Blei, 2012).

4.1. Topic Modelling in a nutshell

The statistical techniques behind topic modelling make a number of assumption that can be summarised as follows: a document is essentially a set (or bag) of words; a document expresses a number of topics of varying importance according to a specific distribution; a topic is expressed with words according to a specific distribution. Thus, by observing a collection of documents, one can empirically estimate the two distributions (document-topic and topic-words) that fit the observed frequencies of words in documents. The basic version of topic modeling details this crudely defined method by selecting a well suited distribution (Dirichlet, hence "Latent Dirichlet Attribution" the name of the most widely used version of topic modeling) as well as the algorithms that can estimate the actual parameters.

From a practical point of view, given a collection of documents (essentially their decomposition as bags of words), a fixed number T of topics and a few hyper-parameters, a topic modeling session produces two matrices.

The first one is a document-topic matrix in which each document is described as a vector across the T topics. In other words, it tells us what topics are the most important ones for each document. This information can be used as such for indexing and comparing document within a smaller vector space.

The second matrix is a topic-word matrix in which each of the T topics is represented as weights associated to each word. In other terms, it gives the words most frequently associated to each topic. This information can be used

to interpret the topics and enable a user to get a readable description of a document in terms of topics.

4.2. Experiment with ASRS data

The experiments we performed on safety reports were designed to answer the following questions:

- Is topic modelling suitable to the nature of our data?
- Are the identified topics relevant to our needs?
- Do these topics actually capture new interesting aspects of events?

The following experiments details are as follows, although they are presented more thoroughly in (Ribeiro, 2014). We used a collection of 167,350 documents from the ASRS database (from 1987 to 2012), and extracted the narrative parts for a total of 17 million words. We used the TreeTagger part-of-speech tagger to use word lemmas instead of wordforms and to remove function words (prepositions, determiners, numbers, etc.). In order to deal with the language variation in the history of ASRS (as described in Section 2.2), all technical words were replaced with their standard acronym (ACFT, WX, etc.). Finally, all tokens were folded to lowercase.

Topic models were computed using the Gensim library (Řehůřek & Sojka, 2010) using the standard method[25] (Gibbs sampling) with a target number of topics $T = 50$. Calculation takes about 2 hours on a 4-core 3.1GHz processor computer.

Although this method is non-deterministic, we could observe through several runs that the results are quite stable, as it has already been observed for corpora this size. The choice of 50 topics is arbitrary, but was finally chosen as the number for which interpretation of the resulting topics was the most satisfactory(see Section 4.4): we will now come to this crucial phase.

4.3. Interpreting the topics

As explained before, a topic model for a given corpus consists in two matrices, document×topic and topic×words. The "Main terms" column of information shown in Table 5 comes from the topic×words matrix. This column contains, for 5 sample topics[26], the 15 words that have the highest probability of expressing it according to the Dirichlet distribution estimated from the observed word distribution. This information is traditionally used for describing a topic to a user and used for testing the relevance and cohesion of this representation (Chang et al., 2009).

A safety expert was presented the 15 most contributing words for each of the 50 topics, and was asked to describe in a few words what each of these topics

[25]The hyper-parameters were left to their default value: $\alpha = 1/T$, $\beta = 1/T$, 50 passes.

[26]The topics' order is insignificant as it is an artefact of the randomisation process at the beginning of the modelling process.

#	Main terms	Expert	Metadata (R)
1	*rwy, txwy, taxi, hold, short, gnd, twr, clr, acft, tkof, line, clrnc, ctl, cross, pos*	Ground	anomaly:ground incursion (0.65); phase:taxi (0.65)
2	*day, hr, time, trip, crew, duty, flt, night, fatigue, rest, leg, fly, min, morning, late*	Fatigue	anomaly:company policy (0.11)
3	*pax, flt, attendant, cabin, smoke, capt, cockpit, seat, back, crew, acft, emer, told, smell, lndg*	Cabin	anomaly:flight deck/cabin (0.60)
4	*wx, ice, turb, flt, tstm, moderate, rain, icing, acft, severe, radar, area, light, encounter, condition*	Weather	primary problem:weather (0.45); anomaly:inflight event (0.37), component:weather radar (0.12)
5	*acft, checklist, flt, call, capt, maint, lndg, make, l, fo, flap, time, control,return, continue*	???	primary problem:aircraft (0.24); anomaly:equipment (0.24); detector:flight attendant (0.23); component:turbine(0.13); component:flap control (0.13)... (6 more)

Table 5: The 5 first topics extracted from the ASRS corpus

could mean. His feedback is presented in the "Expert" column of Table 5. For 43 topics out of 50 the expert was able to identify a theme or a small set of themes that could be expressed by the words with the highest probability values. Although some of the words may seem opaque to a layman, most of them are in fact quite transparent. Contributing words for topic 4, for example comprise both the overal category (*WX* is the standard acronym for *weather*), various meteorological phenomena (*ice*/*icing*, *rain*, *thunderstorm* (*TSTM*)), common modifiers (*light*, *moderate*, *severe*) or consequences (*turbulences* (*TURB*)); all this makes it an easily interpretable topic. This is not the case for topic #5, where no coherence could be found, as the most contributing words are scattered across several aspects of flying an airplane.

The document×topic matrix provides us with another means for interpreting the topics: each document is represented by a vector of weights across the 50 topics. That means that each topic can be viewed as a distribution over the documents, and as such can be compared to the documents' metadata (as described in § 2.2). We thus computed Pearson's correlation coefficient between each topic and each metadata value across the documents (considering 1 if the document's metadata contain this value, and 0 otherwise). This gave us a different, more objective angle to interpret each topic, as we could identify which metadata value was the most strongly associated to each topic. These values are indicated in the "Metadata" column of Table 5, along with the correlation

coefficient's score[27].

First, we can see that for some topics (number 1, 3 and 4 in our selection) one or two highly correlated values (> 0.4) can be identified, and that these confirm the expert's interpretation. Other attributes can appear as secondary correlates, such as flight phase and reporting person, but nevertheless it appears that such topics have captured a well-known aspect of incident reports. This is the case for 38 of the 50 topics. It has to be noted that any aspect of a report can be thus "captured" by a topic. For example, one particular topic was associated to flights in California, the contributing words being the names of locations in this traffic-dense area.

A second case is that of the topics that could easily be identified by the expert but do not show any marked correlation with the metadata. This is the case for topic 2 in our selection, where the only correlated attribute is the company policy, although with a very low score. This kind of topic is extremely interesting, as it shows that corpus analysis by this kind of method can make some aspects of incident reports emerge. Only 2 of these could be identified in the 50 topics examined in our experiment: *fatigue* and *flight planning*[28]. It is important to note that the *fatigue* attribute was added to the ASRS taxonomy, along with other human factors, in 2009. Even though the subset it covers is too small for meaningful results, and is heavily biased because of this temporal constraint, partial analysis indicates that this topic is highly correlated to this attribute.

The 10 remaining topics could not be associated to any single aspect of reports. This is the case for topic 5 in our selection, where the correlated attributes are numerous and scattered, making no more sense to the expert than the contributing words. Other configurations in this category are topics for which several identifiable topics are mixed together.

4.4. A mitigated success

Although we only performed a limited number of experiments with topic modelling on incident reports, we can outline answers to our initial questions.

It appears that topic modelling is very suitable for our data. It is a very robust method that takes clear advantage of large collection of redundant documents as it is the case for incident reports. Most of the topics identified are in fact relevant aspects of these documents, as can be seen through an expert's interpretation. However, only a small fraction of identified topics are both relevant and independent from the metadata attributes, and as such provide an added value.

[27] Only the attributes with a positive correlation higher than 0.1 are presented. This threshold was chosen arbitrarily as the population is too large to have non-significative correlations scores.

[28] This topic more precisely concerns documents where the pilot evokes the flight preparation regarding available fuel, departure time or alternate routes, such as in the report presented in Figure 3.

One of the main limitations of this approach is the granularity of the extracted topics, especially when it is compared to the level of details attained in the organised description and indexing of aviation incident reports. As seen in the previous analysis of the resulting topics, most of the topics do little less than confirm an organisation that is clearly expressed by some of the metadata. If in some cases this method can identify non-encoded aspects, they are difficult to detect among other unavoidably noisy topics. However, this technique can be extremely valuable for reports database that are not supported by a thorough classification scheme and extensive metadata. This can be the case of databases that need to be consolidated, or even for the replacement of an unsuitable taxonomy.

On the technical level, topic models are somewhat sensible to a number of parameters, the first of which is the requested number of topics. We performed several tests on the same data with $T = 10$, $T = 100$ and $T = 200$. None of the topics among the 10 were interpretable, as they all mingle several aspects of the reports. Interesting things happened with 100 topics, including the clear and expected separation of topics (from the 50 described above) that could be identified as an agglomeration of quite distinct sub-topics by the expert. However, this led to only a few such improvements, most other topics were deemed unnecessarily split. With the highest tested value (200), many resulting topics were related to geography, with high-weighted tokens corresponding to airports, beacon codes and city names (mostly in the US). Although these topics were coherent and easily interpreted, their informational value seems quite low. Finally, we could identify a few very stable topics across the variation on T; this is the case for topic 2 (related to fatigue) that was found almost identical in all experiments with $T \geqslant 50$. In the end, the optimal value for T cannot be evaluated without a complete and thorough interpretation of resulting topics, and is estimated to be highly dependent on the collection of documents.

We found few similar studies where topic models were used to analyse and/or process incident reports. Pereira et al. (2013) have used topic modelling in order to estimate the duration of a road traffic incident based partly on the text of the first notification. Although their interpretation of the extracted topics is minimal, they get good results mixing metadata and topics, the latter being in general good predictors of the incident duration, although they did not compare this approach to a more traditional word vector space method.

This experiment confirms, along with the classifier described previously, that most important aspects of incident reports can be captured by the narratives. The next two sections focus on helping the user efficiently make use of these texts to efficiently browse and query the database, with or without the help of categorical metadata.

5. Identifying similar reports with the *timePlot* system

In this section we describe another tool that has been successfully implemented and is currently in use by safety managers for browsing incident reports databases. The *timePlot* search engine retrieves reports that are similar to a source report and displays them along a temporal axis for easier visualisation. We first present the original need that underlies the development of *timePlot*, then describe the tool itself and finally we discuss its limitations and introduce the next generation of systems.

5.1. The need for report similarity

Johnson (2003, p.735) gives the following reason that motivated the development of computer system for managing an incident report database (our emphasis):

> *Identifying trends.* Databases can be placed on-line so that investigators and safety managers can find out whether or not *a particular incident* forms part of a more complex pattern of failure. This does not simply rely upon identifying similar causes of adverse occurrences and near misses. *Patterns may also be seen in the mitigating factors* that prevent an incident developing into a more serious failure. This is important if, for example, safety managers and regulators were to take action to strengthen the defences against future accidents.

Discussions with the safety managers and analysts from several companies and regulation authorities confirmed Johnson's stress on the importance of browsing a database while looking for similarities. This concept is essential to the discovery of recurring events that need to be avoided. More precisely, the scenario we address in this section is the following: given an already identified occurrence (and its report), can we quickly and easily find other occurrences in the database that share the same characteristics? The similar features can be of any nature: time, place, type of aircraft, weather, flight conditions, problem encountered, actions taken, results, etc.

The need for identifying similar occurrences is manifested in two types of situations. The first one is the monitoring of the incoming data flow. Whenever a report arrives, the initial receivers may want to verify if this incident is an isolated occurrence or is part of a larger trend. If it is part of a trend, he would want to estimate the frequency of the events in question, judge the risk they represent and, if necessary, take corrective actions.

The second situation is when a decision is made to investigate into a particular issue or risky scenario. In such a case the experts need a large body of examples covering all possible aspects for a qualitative analysis. In this situation, one way to approach the problem is to identify (usually from their memory and intricate knowledge of the database) a particular prototypical occurrence and then use it as a query to search the database for similar reports.

Take the example in Figure 3. In an ideal ECCAIRS world all relevant aspects of the occurrence will be coded in the metadata and all other occurrences in the database will also have coherently coded values. A user would then be able to use this report as a source and query the database looking for reports that share all or most of its characteristics. He will find many reports where there were deviations to an alternate airport. Quite a large subset will probably concern occurrences where extensive delays (holding) lead to burning a lot of fuel and hence to the decision to divert. A larger-than-normal subset of those might concern a diversion from some other airport—i.e. not the airport given in the example in Figure 3. The expert would not have thought of looking at that particular airport in the first place, but now will find it interesting and would want to investigate further. He might find out that the cases concerning that airport almost always implicated bad weather conditions. A pattern is identified. If this is the case, it may lead to changing minimal fuel requirements for this destination or training the crews to better prepare for the probability of diversion if bad weather is forecasted for that destination.

All of the above-mentioned bits of information are also present in the narrative. Given that, as we saw, metadata is not always adequate (or available) we built a system that relies only on the report narratives to identify similar reports. The main advantage of using narratives is their availability and coverage: there are always free text narratives associated with incident reports and they contain all of the information expressed by the reporter. The downside is that, although present in the narrative, the information is less structured and, moreover is very noisy. In the above-mentioned example, the reporter tells of a wet runway at the destination airport. This information is not of primary importance for the occurrence and an expert coder would probably not have included it in the metadata. A narrative-based similarity system however will consider wet runways as a feature for similarity.

When designing the text-based similarity system we took into account this trade-off and made use of interactive visualisation technique, allowing quick and easy access to the results but at the same time explaining *why* the results are as they are. For this reason we privileged straightforward and simple linguistic processing as it means less opaque treatment, and is directly understandable by the users. Another (typical) trade-off when designing information retrieval systems is finding the precision/recall sweet spot. Following the transparency principle we decided for a high-recall strategy coupled with easy filtering. The initial results are noisy but the users have the possibility of further refining and filtering them with a few mouse clicks. Again, this follows the initial demand where the expert should be able to refine his information need little by little as he analyses the results. In the hypothetical scenario described above, the fact that a pattern was identified for a totally unrelated airport depended at first on a loose interpretation of similarity. Such a use pattern is typical in the processes of discovering trends and making connections when performing safety investigations.

5.2. Overview of the timePlot *system*

The general principle of the *timePlot* system is straightforward. For a given report (the *source* report) the tool identifies similar reports among the ones that are indexed in its database. The reports are presented chronologically on a two-dimensional scatterplot. Each point represents a report: the higher a point is, the more similar the report it represents is with the source report.

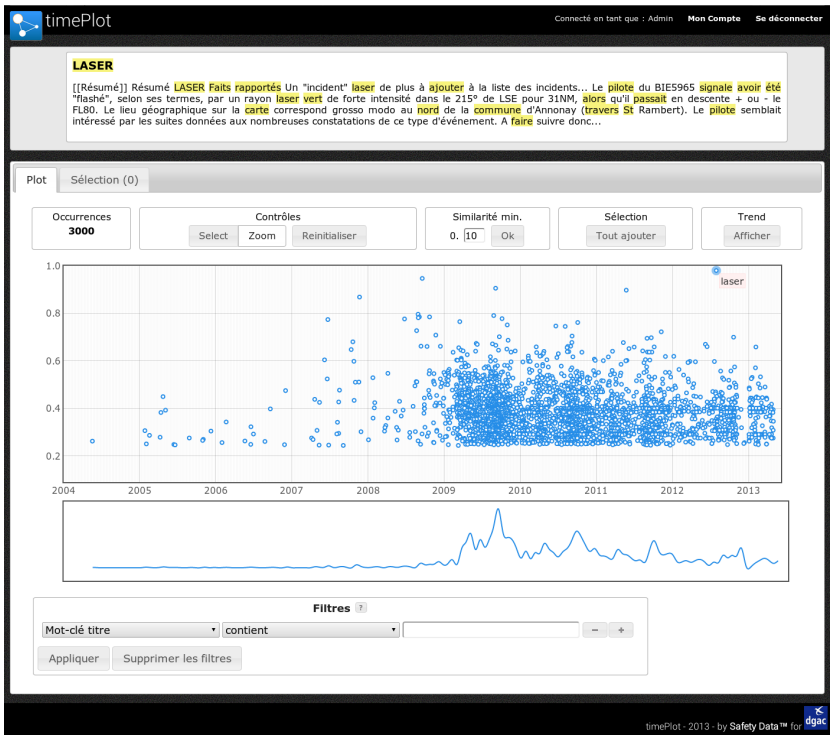

Figure 4: *timePlot* user interface

On Figure 4 we see snapshot of the *timePlot* interface when applied to the DGAC database of French reports described in Section 3. The source report concerns a laser pointer incident report as source and many similar incidents are displayed on the scatterplot underneath. The graphical disposition of the points on the scatterplot allows the user to instantly identify this particular incident type[29] as a trending one, as most of the similar reports are concentrated towards the right-hand side of the plot.

[29] In 2008, relatively cheap and extremely powerful laser pointers became available for purchase online and in some specialised stores. For some reason some people started pointing them at approaching aircrafts, creating a serious safety hazard. Of course, no metadata enables the coding of this specific problem.

Similarity with the source is represented by the vertical position of the points on the plot. The ones high on the plot share many of the source's characteristics. Point lower on the graph are less similar, and thus potentially irrelevant. The user can quickly verify the precise words shared by any two reports by hovering with the mouse on the corresponding point. As is visible on Figure 4 the shared words are highlighted in the text of the source report. Hovering on a point also presents the title of the corresponding document and clicking on the point opens a dialog window containing the entire document.

The filter bar on the lower part of the screen allows the user to filter out reports based on keywords and metadata, and thus to focus on a subset of the retrieved reports. In addition, the user can select a similar report and make it the new source report, thus interactively exploring the database in an hypertextual way.

These features are all intended to facilitate the navigation within the set of similar reports. By minimising the effort the users have to make in order to understand a given output of the system we can intentionally allow for a noisier output and higher recall. Instead of "being forced to continually navigate 'another 10 hits' to slowly identify relevant reports" (Johnson, 2002), users can examine the graph and, by hovering on several points and looking at the highlighted terms, they get a good enough idea on the composition of the results and further formulate their search strategy.

5.3. Calculating Similarity

The core of the similarity calculation is also straightforward. Given any pair of documents, the system produces a similarity score, between 0 and 1 representing the relatedness of the documents. The score is based on the *lexical overlap* of the narrative parts of the two documents. The more words they share in common the more similar the documents are. This is a classical implementation of the vector-space Information Retrieval principles (Manning et al., 2008), although the similarity is calculated between documents and not between a query and a document.

The details of the processing chain is as follows. First the texts are *tokenized* and *stemmed*, and a stoplist is applied to select the terms in each report. The first processing stages are identical to the processing described in Section 3.

Each document is represented by a vector where each dimension corresponding to a term in the collection, and each value is the relative weight if this term in the document. We used the classical TF*IDF measure, that takes into account both the numbers of occurrence of this term in the document and the rarity of this term in the collection.

Finally two documents are compared by computing the cosine (or dot product) between the vectors that represent them.

This classical approach to similarity is quite rudimentary in regard to recent development in IR. It considers each document as a bag of words (without taking word order into account) and does not make use of term similarity (synonymy or other semantic relationships) either through specific linguistic resources nor

unsupervised statistical methods. This has been our choice for a practical reason, which is to keep the similarity as transparent to the user as possible. Also, this word-level similarity between reports allows a greater interaction with the user, such as word filtering.

5.4. TimePlot *in use*

TimePlot has been proposed to aviation safety experts at both the national (France) and European level. It is currently in active use in the French DGAC and in advanced testing in a French airline company's safety intelligence service awaiting integration in their safety management system.

A standard evaluation of this tool along the NLP standards, i.e. running it on a gold standard data and assessing its efficiency in terms of precision and recall, has not been performed as the task itself does not comply with the requirements of such an approach.

Instead, we have been observing its use both quantitatively through automated logs and qualitatively through user experience and feedback.

At the DGAC, where the tool is at a most mature stage, there are currently 136 active users. Data is synchronised with their ECCAIRS database on a monthly basis. We have had a largely positive feedback from the users and the DGAC are also starting an occurrence data sharing programme supported by the tool. The operators (airports and companies) willing to share part of their incident data will get free access to the tool with all the data that other operators participating in the programme have shared. Currently 5000 queries are performed yearly at the DGAC alone.

One interesting scenario concerns the airline's testing of the tool. As part of the test we had provided the tool loaded with a database of publicly available incident reports. One of the questions that the safety officers were interested in concerned events that occurred at some of their diversion[30] airports. For one particular airport in central Russia, the tool identified a larger than normal concentration of runway overruns—cases where the landing aircraft did not manage to stop in time. The problem was related to improper drainage of the runway surface and the company updated the procedures for landing there in case of emergency according to these findings.

In another case the experts were asked to investigate a series of specific incidents. The identification of similar incidents over an extended time period allowed them to determine that the original cluster was "a statistical accident" and not a developping trend, thus avoiding the (very coslty) creation of a special investigative task force.

Another success story is related to regulation about the use of mobile phones on airplanes, which changed recently and led the company to consider allowing

[30] Airports to be used in case of an emergency. Having accurate and up to date information about these airports is problematic for companies, as they do not use them during normal operations. At the same time, the need for such information is of paramount importance when performing an emergency landing.

their use in the cockpit by the pilots. Using the tool, they searched for reports about possible interferences and found one case where a mobile phone of a passenger seated in one of the front rows interfered with crucial instruments. Based on this, it was decided to maintain the ban in the company's standard operating procedure.

5.5. The next step: targeted search

By monitoring the actual use of the *timePlot* system by its intended audience, we found that in some situations the system was used well beyond its designed limits.

Let us recall that the system is intended to help the identification of similar occurrences; after processing, the user can tackle the initial noise by filtering the results using a combination of keywords and metadata fields. In a later version we introduced[31] the possibility fo a user to manually enter or copy/paste a report narrative and the system would compute its similarity with stored reports.

We noticed that at some cases this functionality was not used to find a scenario, but rather to model a certain aspect of an incident. Users would input a variety of semantically related words or variants in the full-text field essentially using the system as a (crude) full-text search engine. After calculation, the users would scan the scatterplot, identify reports not matching their initial need and try to filter them out with keywords and Boolean operators.

A user would, for example enter *fatigue*, *tired*, *rest*, and *sleep* in the full-text field. In this example the user tries to identify reports where fatigue was a factor. Afterwards, when looking at the results the user would notice that some reports mention "*metal fatigue*[32]" and then apply a filter excluding the word *metal* from the results. The realisation that one of the initial terms (*fatigue*) is ambiguous, and that searching for it yields irrelevant results would come when looking at the results after the first iteration and not be expressed in the initial query. This type of narrowing down of the search criteria and progressive specification of the information need through query reformulation is typical for modern information seeking strategies (Jansen et al., 2009).

This example shows how a clear understanding of both the tools and the data they manipulate allows the users to devise more intricate strategies to satisfy a given need for information. The *timePlot* tool, not being designed with such a use in mind naturally does not yield optimal results. However the fact that it was used in such a way clearly indicated that such needs must be addressed with a purpose-built system. More importantly, it indicated that the users are willing to engage in such an iterative information seeking process with multiple "round trips" from search to data.

Such behaviour from the users is understandable in the sense that the information they seek is ever more elusive. The term "*non-technical signal*" came

[31] This possibility was introduced for purely technical reasons. The initial update cycle of the application was too slow and at some cases the users didn't want to wait before checking for similar reports.

[32] *Fatigue* is used to denote the weakening of a material under forces.

about in one of our discussions, making a distinction between *technical* matters that are clearly identifiable with simple terms (such as the names of specific components) and *non-technical* matters, such as human factors issues where key-word approaches are not powerful enough to reflect complex issues such as *confusion* or *distraction*.

In the end, whereas modern search engines put the emphasis on precise results with minimal engagement, we observed that in the context of searching for complex issues in noisy textual data, a human could not be expected to produce a coherent enough query *ex nihilo*. The tools, rather than simply aiming at the best possible result, should let the user "build a relationship" with the data they manipulate. After a first (underspecified) query, the tool ideally would give a picture of both signal and noise and allow the user to build on that impression to further refine their need and adapt to the underlying reality of the data.

In the following section we present an approach to capturing non-technical signals using *active learning*, which, by keeping the user in the loop, allows for such a progressive refinement of the information need.

6. Interaction and active learning

We described in the previous section how the *timePlot* tool was (mis-)used to model a facet of an incident, rather than to look for similar incidents. This usage scenario led us to design an approach that relies on the availability of an expert and use a variant of machine learning techniques: active learning (Olsson, 2009). These variants are based on traditional supervised learning methods, but take into account the fact that training data are expensive to get when an expert is required for labelling items. Active learning strategies try to make a smart usage of the expert's time by submitting to his judgment only the difficult or borderline items. This can only be done through an iterative process with a dose of interaction with the user.

In this section we describe the intended approach, the algorithm we designed and a simulation we are currently running as part of *AnonymisedCompany*'s R&D program in order to better understand the behaviour of the active learning approach and tune it before we submit it to real users.

6.1. An interactive approach to signal detection

The system we present here is based on the observations of the use of the *timePlot* tool and on the successful performance of the machine learning approach described in Section 3. The basic idea behind the system is to allow the users to model a given aspect of an incident by providing examples of documents that are related to the particular aspect. We start with the assumption that the aspect is partially identifiable by a query using a full text search engine and/or available metadata. A user interested in confused flight crews will presumably start by querying the system for documents containing the word "confusion". This set will, however contain some documents that do not match the user's

information need[33]. When looking through the documents, the user will notice this and would like to exclude them from the search. At the same time there will be documents that do not contain the word "confusion" and that are relevant. The system should also be able to identify such documents.

We have systematised this process into what we have call "creating a *Dimension*". A *Dimension*, from a user's point of view is a dynamically created label that can potentially apply to any report in the corpus, as well as any new report introduced. Conceptually, creating a *Dimension* can be compared to introducing a new metadata attribute / value pair to an existing taxonomy. However, the difference is that we seek to render the process the least time-consuming as possible and not require extensive coding of all the existing reports.

From a system's point of view a *Dimension* is no more than a classifier that produces a yes/no partitioning of the corpus. The algorithm described in the next section shows the process for creating and training this classifier.

6.2. Active learning algorithm

The outline of our system is the following: we start with a rough estimation of what the expert considers as the target (positive) reports. We train a classifier based on this data, and then apply it to the entire collection. Due to the nature of classification algorithms (and their need for generalisation), this classifier provides a different set of positive reports. Using the error margin (or probabilistic confidence score) provided by the classifier, we can identify borderline reports, on both sides of the decision: we select these few fairly positive and fairly negative items and submit them to the expert's judgement. Based on his decisions, we obtain a new approximation of his needs, and can train another classifier, and so on until the expert reaches a satisfactory result. This active learning principle is also called uncertainty-based sampling and has been proposed in a number of NLP tasks such as information extraction (Kristjannson et al., 2004) and semantic role labelling (Roth & Small, 2006), among others.

Algorithm 6.1 shows in details the active learning algorithm for training a Dimension. Given a corpus $\mathcal{C}$ of safety reports, we wish to calculate a dimension vector $\mathcal{D}$ assigning a dimension yes/no value to each document in the corpus. The expert kicks the system off by providing an initial approximate set of positive examples $\mathcal{P}$. These are either the result of a keyword search for keywords highly suggestive of the target dimension, a set of similar reports identified with *timePlot* or a handful of manually selected documents. The system also requires a set of training parameters which depend on the classifier used (e.g. C and ϵ for a linear SVM classifier), and a set of training features $\mathcal{F}$ to represent the textual content of the reports (see § 3). The final input parameters are the "bootstrap" threshold t, giving the minimal distance from the SVM hyperplane

[33]Consider documents speaking for confusing call signs, for example. XX259 and XX299 flying at the same time in the same area makes it rather hard to communicate with ATC over the radio but does not necessarily amount to the flight crews being confused about what they are supposed to do.

Input: corpus $\mathcal{C}$, initial positive set $\mathcal{P}$, training parameters, feature set $\mathcal{F}$, bootstrap threshold t, review count n
Output: dimension vector $\mathcal{D}_n$

$\mathcal{T} \leftarrow$ new training set;
$\mathcal{T}.p \leftarrow \mathcal{P}$;
$\mathcal{T}.\mathcal{P} \leftarrow \varnothing$;
$\mathcal{T}.\mathcal{N} \leftarrow \varnothing$;
$i \leftarrow 0$;
repeat
  `// Train model`
  $\mathcal{T}.n \leftarrow$ random sample with cardinality $(|\mathcal{T}.\mathcal{P}| + |\mathcal{T}.p|) - |\mathcal{T}.\mathcal{N}|$;
  Train model $\mathcal{M}_i$ using $\mathcal{T}.\mathcal{P} \cup \mathcal{T}.p$ as positive and $\mathcal{T}.\mathcal{N} \cup \mathcal{T}.n$ as negative examples and $\mathcal{F}$ as features;
  `// Calculate dimension vector`
  $\mathcal{T}.p \leftarrow \varnothing$;
  $\mathcal{D}_i \leftarrow$ new dimension vector;
  foreach *doc* $\in \mathcal{C}$ *indexed by* j **do**
    $\mathcal{D}_i$[j] $\leftarrow$ apply $\mathcal{M}_i$ to doc_j using $\mathcal{F}$;
    `// Bootstrap calculated positives`
    if $\mathcal{D}_i[j] > t$ **then** add doc_j to $\mathcal{T}.p$;
  end
  `// Review marginal documents closest to hyperplane`
  for n *positive and* n *negative docs* $\notin \mathcal{T}.\mathcal{P} \cup \mathcal{T}.\mathcal{N}$ *closest to hyperplane* **do**
    Expert reviews doc_j;
    Expert adds doc_j to $\mathcal{T}.\mathcal{P}$ or $\mathcal{T}.\mathcal{N}$;
  end
  $i \leftarrow i + 1$;
until *expert satisfied*;
return $\mathcal{D}_i$;

Algorithm 6.1: Iterative dimension training

for a document to be included in the positive set on the next iteration, and the review count n giving the number of documents to be reviewed at the end of each iteration in the margin of the SVM hyperplane.

The training set $\mathcal{T}$ is comprised of four sets of documents: $\mathcal{T}.\mathcal{P}$, the real positives which have already been reviewed by the expert, $\mathcal{T}.\mathcal{N}$, the real negatives which have already been reviewed by the expert, $\mathcal{T}.p$, the positives automatically calculated by the previous model above the bootstrap threshold (initially provided by the expert in $\mathcal{P}$), and $\mathcal{T}.n$ a random sample of documents assumed to be negative, with a cardinality to balance the positive and negative examples. It is of course possible (and desirable) to give reviewed positives and negatives a higher weight than calculated positives/random negatives.

At each iteration, the system first trains a new model $\mathcal{M}_i$ given the current

training set $\mathcal{T}$. It then calculates a new dimension vector $\mathcal{D}_i$ using the model $\mathcal{M}_i$. Within the algorithm, we'll assume $\mathcal{D}$ contains a real positive or negative distance from the SVM hyperplane, although it is trivial to convert this to a yes/no answer by taking positives to be yes and negatives to be no. Finally, the system reconstructs $\mathcal{T}$ as follows: $\mathcal{T}.p$ is automatically calculated by taking all documents where the distance from the hyperplane exceeds the bootstrap threshold. The expert is then asked to review the n documents closest to the hyperplane margin on both sides, and determine whether they are really positives or negatives, assigning them respectively to $\mathcal{T}.\mathcal{P}$ or $\mathcal{T}.\mathcal{N}$. The assumption is that correctly reclassifying a small number of documents in these marginal areas allows us to converge much more quickly than a random review of documents.

The learning ends when the expert is satisfied with the dimension values assigned to documents—presumably when the hyperplane correctly distinguishes the majority of documents reviewed.

6.3. Simulation and discussion

In order to better understand the behaviour of the system and assess its usefulness, we ran several simulations using existing metadata as a validation criterion, substituting itself to the expert's judgement. At each iteration, reclassifying the documents from the marginal areas is done based on whether they are *true* positives or negatives for the target metadata attribute.

We used the same classifier and feature set as described in Section 3, i.e. a linear SVM based on stems and stems n-grams. For an estimation of the classifier's margin we used the probabilites provided by the libLinear library, which are based on the distance between an item and the trained model's hyperplane. The bootstrap threshold t is set at 0.8.

As a metric of performance, at each iteration we measured precision, recall and F1 scores for overlap between the documents identified by the system and the documents classified according to the target metadata attribute.

Table 6 shows the results of the simulation on a subset of the French DGAC corpus consisting of 44,191 documents (arbitrarily selected on a temporal criterion from the whole corpus described in Section 3). The task consists of creating a *Dimension* for bird strikes. The initial set $\mathcal{T}.p$ contains all documents that contain the word "oiseau"[34]. We have set the review count n to 10, meaning that at each iteration the 10 positive and 10 negative items with the lowest margins are submitted to the expert (or here, have their status revised according to their metadata).

The first row of Table 6 shows the state of the system at query-time. The query has returned 1,534 documents. From those, 1,413 are considered *true* positives (have the occurrence category *BIRD*). The query is quite precise, with a precision of 92.11%, but its recall is 43.85%, meaning that less than half of the documents categorised as *BIRD* contain the word "oiseau" (in fact, most reports signal the exact species of bird encountered).

[34] "bird" in French.

The second row shows the state of the system after a model has been trained on the initial set. $\mathcal{T}.p$ now contains the documents classified by the model. While no "human" reclassification has yet been performed, 346 new true positive documents have already been correctly identified by the system.

The subsequent rows show the state at each iteration. At iteration 3, for example the expert has reclassified a total of 40 documents (28 as positives and 12 as negatives). After the corresponding retraining, the system identifies 176 more true positives as compared to the state at iteration 1. We can see that the F1 score is steadily increasing with each iteration, illustrating how expert input on a small amount of documents iteratively refines and tunes the classifier.

i	$\mathcal{T}.p$	$\mathcal{T}.\mathcal{P}$	$\mathcal{T}.\mathcal{N}$	**True+**	**P (%)**	**R (%)**	**F1 (%)**
0	1534	0	0	1413	92.11	43.85	59.42
1	1957	0	0	1759	89.88	54.59	67.93
2	2035	17	3	1804	88.65	55.99	68.63
3	2205	28	12	1935	87.76	60.06	71.31
4	2379	41	19	2104	88.44	65.30	75.13
10	3347	140	40	2877	85.96	89.29	87.59

Table 6: Results for bird strikes (DGAC corpus)

Table 7 shows the results of another simulation, this time on 7,025 documents from the American ASRS database (selected on a temporal criterion from the corpus described in Section 4). We simulated the search for incident reports where *confusion* was a factor and we use the *Human Factors* attribute of the *Person* entity as a validation criterion. We tested for those documents classified with the value *Confusion*. The initial query is the word "confusion".

While this configuration is closer to the real-word use the system is intended for, it is also a much more difficult task than identifying bird strikes. This difficulty can be estimated by training a simple classifier for this metadata: our best configuration achieved only 66% F1-score, while we saw in Section 3 that we could reach 95% for the *BIRD* category in the DGAC corpus.

Accordingly, the system performance is worse than in the previous scenario, but the behaviour is comparable. At iteration 3 the system has identified 253 more true positive documents with only 40 being submitted to the expert for validation. After 10 iterations, even though the F1 score is still below 50%, recall has doubled.

Globally these results are encouraging. They demonstrate that it is possible to better capitalise on the expert's time and, with this type of active iterative process, effectively "propagate" the judgement to a large proportion of the documents. While validating the general principle, these experiments also pose a number of questions. The most important one currently is the relationship between the initial query and the output of the system. We have observed that the system behaves differently depending on both the precision and the recall of the query. We also observed that, depending on the query, varying parameters such as the bootstrap threshold, the review count and the additional weight given

i	$\mathcal{T}.p$	$\mathcal{T}.\mathcal{P}$	$\mathcal{T}.\mathcal{N}$	**True+**	**P (%)**	**R (%)**	**F1 (%)**
0	774	0	0	472	60.98	25.46	35.92
1	1048	0	0	574	54.77	30.96	39.56
2	1280	14	6	670	52.34	36.14	42.76
3	1443	24	16	725	50.24	39.10	43.98
4	1564	26	34	765	48.91	41.24	44.74
10	1936	57	123	900	46.49	48.54	47.49

Table 7: Results for *confusion* (ASRS corpus)

to documents already reviewed have different effects and can greatly improve performance. As we can not have control on the query itself, we are searching for methods to automatically determine the optimal values of these parameters. We are also looking forward to building the graphical user interface and proposing the system to real word users. This will allow for much more realistic testing as we will be able to directly measure performance based on the proportion of yes/no judgements at each iteration.

7. Conclusion

The work presented in here, in addition to providing an operational solution to identified safety needs, addresses a number of more general issues.

First, the problems faced by experts attempting to analyze a large quantity of textual data in order to find emerging dangers and risks are present in a large variety of industrial contexts: energy (power plants, oil and gas extraction), transportation (railway, buses, rapid transit systems), heavy industry (chemistry, founderies, manufactures), health, etc. Each domain of activity and individual structure (company, state or international regulation organisation) is different in terms of volume, reports origins, textual characteristics, etc. Although common methods and techniques can be identified, that fact remains that specific approaches have to be followed for each situation. Nevertheless, aviation safety is seen as the field where the most advanced incident reporting systems have been developed, and has been taken as an explicit example by varied studies.

These solutions, developed in such a resource-rich and advanced environment, can now be redesigned and adapted to more virgin domains. Certain techniques might be particularly applicable to such domains. Indeed, while the lack of metadata and training material can be an obstacle for automatic classification with supervised techniques, this state of affairs encourages us to deploy unsupervised techniques such as topic modelling in order to clear the ground and get a first global vision of the main tendencies expressed by data. In some cases, more generic methods and tools can be directly transferred.

This is the case of an application that *CFH / Safety Data* has started in the domain of *nursing homes*, for which most of the problems and solutions mentioned in this article are relevant. These institutions produce incident reports and categorize incidents according to a taxonomy of activities (drug administration, nursing, laundry, etc.). As for aviation, these reports are emitted by a range of personnel, are written in a technical style and contain a large number of acronyms. Although the taxonomy is oriented towards technical issues, most of the safety managers' concerns are aimed at orthogonal aspects such as human factors and arduousness of work. This naturally calls for the identification of specific dimensions such as presented in Section 6. The report system, although it covers a large number of houses, is more flexible than the international standards of aviation: this means that a closer inspection of the reports (e.g; with techniques such as topic modelling presented in Section 4) can be considered in order to modify the taxonomy.

Another domain in which we have already applied similar techniques is the space industry, and more specifically satellite manufacturing. In these procedures where extreme precision is expected, reports are written for each encountered case of non-conformity with the technical requirements. Managers have defined a generic taxonomy to describe these reports, with wide-coverage categories such as severity and causes. Efficient monitoring of this database is now performed through an adapted version of the timePlot tool.

Secondly, in the majority of interesting cases, the target concepts and signals sought by safety experts are not formalised until the problem has been clearly identified.

Contrary to the information retrieval model of "finding a needle in a haystack", in the mentioned cases we do not even always know what a needle looks like. This raises a number of problems, not the least of which being the evaluation of proposed technical approaches. NLP, like other empirical fields, distinguishes intrinsic and extrinsic evaluation (Spärck Jones & Galliers, 1996). Intrinsic evaluation targets the efficiency of the tool in itself, as when we evaluated our classifier, while extrinsic evaluation aims at measuring the efficiency of the tool in its usage environment (in other words, its actual usefulness). This extrinsic part of the evaluation has yet to be performed, and cannot be achieved using traditional NLP evaluation methods of comparing the results with a benchmark. At this stage, extensive usage and user satisfaction are the best indicators we can identify as to the usefulness and relevance of the solutions we propose.

Acknowledgements

The work presented here is the result of more than 10 years of a joint research effort, and it involved many people we want to thank here.

- for *CFH / Safety Data*: Edith Galy, Aleksandar Kalev, Michel Mazeau, Christophe Pimm and Nicolas Ribeiro;

- for *CLLE-ERSS*: Didier Bourigault and Cécile Fabre.

We also want to give a special thanks to Reinhard Menzel (formerly from EASA) and the other aviation safety experts for giving us access to their extensive knowledge.

References

ADREP (2010). *ADREP 2000 taxonomy*. ICAO.

Andréani, V., Fabre, C., Raynal, C., & Tanguy, L. (2013). *Techniques de TAL au service de la constitution d'une base de REX et de son analyse*. Technical Report P10-5 Institut pour la Maîtrise des Risques.

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*, 77–84.

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 22* (pp. 288–296). Curran Associates, Inc.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

DGAC (2013). *Rapport sur la sécurité aérienne 2013*. Technical Report DGAC. URL: `http://www.developpement-durable.gouv.fr/Rapport-sur-la-securite-aerienne.html`.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Hall, D., Jurafsky, D., & Manning, C. D. (2008). Studying the history of ideas using topic models. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 363–371). Stroudsburg, PA.

Helmreich, R. L. (2000). On error management: Lessons from aviation. *British Medical Journal*, *320*, 781–785.

Ho, C.-H., & Lin, C.-J. (2012). Large-scale linear support vector regression. *Journal of Machine Learning Research*, *13*, 3323–3348.

ICAO (2013). *2013 safety report*. Technical Report ICAO.

Jansen, B. J., Booth, D. L., & Spink, A. (2009). Patterns of query reformulation during Web searching. *Journal of the American Society for Information Science and Technology*, *60*, 1358–1371.

Johnson, C. W. (2002). Software tools to support incident reporting in safety-critical systems. *Safety Science*, *40*, 765–780.

Johnson, C. W. (2003). *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting*. University of Glasgow Press.

Kristjannson, T., Culotta, A., Viola, P., & Callum, A. M. (2004). Interactive information extraction with constrained conditional random fields. In *proceeding of the Conference of the American Association for Artificial Intelligence (AAAI)*. San Jose, CA.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Menzel, R. (2004). ICAO safety database strengthened by introduction of new software. *ICAO Journal*, *59*, 19.

Newman, D., Chemudugunta, C., Smyth, P., & Steyvers, M. (2006). Analyzing entities and topics in news articles using statistical topic models. In *Intelligence and Security Informatics* (pp. 93–104). Springer.

Olsson, F. (2009). *A literature survey of active machine learning in the context of natural language processing*. Technical Report Swedish Institute of Computer Science.

Pereira, F. C., Rodrigues, F., & Ben-Akiva, M. (2013). Text analysis in incident duration prediction. *Transportation Research Part C: Emerging Technologies*, *37*, 177 – 192.

Pimm, C., Raynal, C., Tulechki, N., Hermann, E., Caudy, G., & Tanguy, L. (2012). Natural language processing tools for the analysis of incident and accident reports. In *Proceedings of the International Conference on Human-Computer Interaction in Aerospace (HCI-Aero)*. Brussels.

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). Malta.

Ribeiro, N. (2014). *Visualisation interactive de similarité textuelle - Intervention du topic modeling*. Master's thesis Université de Toulouse.

Roth, D., & Small, K. (2006). Margin-based active learning for structured output spaces. In *Proceedings of the European Conference on Machine Learning (ECML)* (p. 413–424).

Spärck Jones, K., & Galliers, J. R. (1996). *Evaluating Natural Language Processing Systems: An Analysis and Review*. Berlin: Springer Verlag.

Stephens, C., Ferrante, O., Olsen, K., & Sood, V. (2008). Standardizing international taxonomies. In *ISASI Forum*.

Urieli, A. (2013). *Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit*. Ph.D. thesis Université de Toulouse II le Mirail.