SafetyRank: a tool for retrieving safety alerts information using natural language processing

Wander Fernandes Júnior
Graduate Program in Applied Computing (PPComp)
Instituto Federal do Espírito Santo (Ifes)
Campus Serra - ES - Brazil
wanderfj@gmail.com

Jefferson Oliveira Andrade
Karin Satie Komati
Kelly Assis de Souza Gazolli
Graduate Program in Applied Computing (PPComp)
Instituto Federal do Espírito Santo (Ifes)
Campus Serra - ES - Brazil

Prepared for Presentation at
American Institute of Chemical Engineers
2020 Regional Meeting
CCPS Latin American Virtual Conference 2020
October 28-29, 2020

AIChE shall not be responsible for statements or opinions contained
in papers or printed in its publications

SafetyRank: a tool for retrieving safety alerts information using natural language processing

Wander Fernandes Júnior
Graduate Program in Applied Computing (PPComp)
Instituto Federal do Espírito Santo (Ifes)
Campus Serra - ES - Brazil
wanderfj@gmail.com

Jefferson Oliveira Andrade
Karin Satie Komati
Kelly Assis de Souza Gazolli
Graduate Program in Applied Computing (PPComp)
Instituto Federal do Espírito Santo (Ifes)
Campus Serra - ES - Brazil

Keywords: Learn from Experience; Process Knowledge Management

Abstract

In industrial areas the issuance of safety alerts after the occurrence of accidents is common. In this context, this work proposes a computational tool that searches and ranks public industrial safety alerts using the history of accidents in the naval and oil and gas industry. An algorithm was developed for reading documents, performing natural language processing (for tokenization, data cleaning and content stemming), indexing and storage in databases, and search and ranking of results. The search results are ordered by similarity with the input data using the Okapi BM25 method. The use of this tool made it possible to search, in a practical and quick way, for previous accident alerts that are relevant to a certain activity.

1 Introduction

According to data from the Digital Observatory on Occupational Health and Safety, from 2012 to 2018 Brazil recorded 4.5 million accidents and 16,455 work-related deaths. In the same period, social security expenses associated with the injured reached R$ 79 billion [1].

In organizations, many accidents occur because those involved do not know how to avoid them, even though others in the same organization have such knowledge. Accidents can occur less frequently if lessons learned are disseminated, discussed and recorded, so that actions taken in the past are accessible to people in similar situations [2].

In industrial areas, it is common for safety alerts to be issued in digital format after accidents have occurred. These text documents contain reports of the occurrence and lessons learned that aim to prevent harm to people and strengthen the safety culture through the sharing of experiences. Figure 1 illustrates alerts issued by the Brazilian National Petroleum Agency (ANP), Chemical Center for Process Safety (CCPS), American Coast Guard and International Association of Drilling Contractors (IADC) obtained from the internet [3, 4, 5, 6]. As there is a large volume of documents produced in a decentralized manner by different organizations, conducting a search on the contents of the documents is not efficient.

Safety Alert 002 - ANP/SSM

Integrity Management of Passive Fire Protection

Passive Fire Protection

Beacon

DEKRA

What if that "wrong" instrument reading is correct?

What Can You Do?

Think about what an "abnormal" instrument reading could mean!

UNITED STATES COAST GUARD

MARINE SAFETY ALERT

Inspections and Compliance Directorate

Don't get burned!
Be ready with a proper Fireman's Outfit

Safety ALERT

IADC

Finger Injury results from BOP Handling

Alert 4-19

WHAT HAPPENED:

Securing plate BOP was landed on

CONTRIBUTING FACTORS:

Inadequate design

Task plan not fit for purpose

Inadequate change management

Inadequate Procedure

People – Hazard Awareness

Figure 1. Examples of safety alerts.

The digitalization of the real world generates a large amount of data that needs to be properly treated in order to be transformed into useful and relevant information [7]. Due to the increasing availability in the number of electronic text documents, work on machine learning applied to texts has gained more space [8].

Natural language processing is the use of human languages, such as English or French, by computers [9]. There is a wide field of applications, such as: sentiment analysis, topic modeling, document classification, information retrieval systems, chatbots, language detection and translation, summarization, text generation and prediction [10].

This work proposes a computer program to search for public industrial safety alerts related to accidents in the naval and oil and gas industry. The developed algorithm performs the reading of accident records, the processing of natural language (for tokenization, data cleaning and content stemming), indexing and storage in databases, in addition to the search for alerts and ranking of results. The search results are ordered by similarity with the input data using the Okapi BM25 function [11].

2 Theoretical framework

In this section, concepts related to organizational learning and information retrieval systems are presented.

2.1 Organizational Learning

In the corporate environment, the term lesson learned means a change in personal or organizational behavior as a result of learning from experience. It is a recommendation (positive or negative) from which other people can learn to improve their performance on a specific task or objective [12].

Although it is more challenging, it is much more beneficial to learn from the experiences of others than to learn from your own mistakes - especially from bad experiences that you would not want to have. By avoiding the repetition of errors and maintaining the execution of the successes obtained, it is possible to obtain a continuous improvement in the performance of an organization [12].

As shown in Figure 2, it is possible to define a quadrant of possible learning approaches.

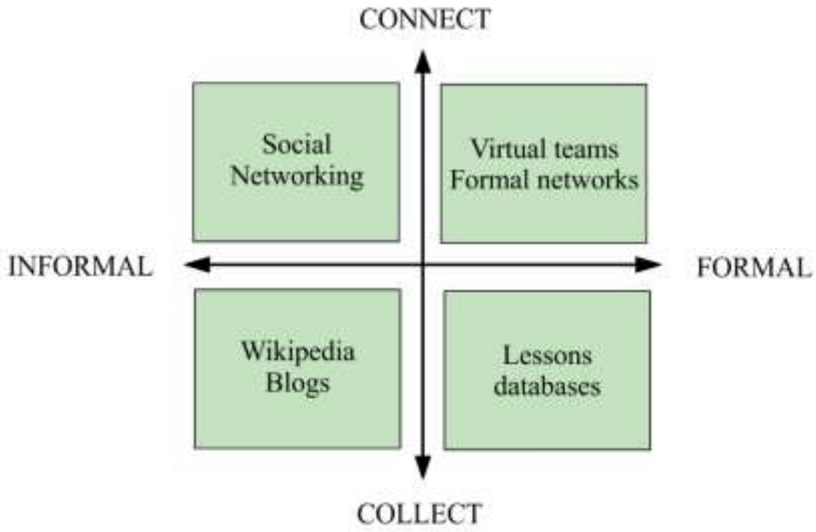


Figure 2. Learning approaches [12].

In the lower left quadrant is the Informal / Collect approach, which includes voluntary community tools, such as Wikipedia. Although easy to use, there is a real risk that crucial lessons will never enter the system [12].

The Formal / Connect approach (upper right quadrant) is characterized by formal networks with specialists who can share knowledge with other participants. The transmission of lessons is done through dialogue. They are ideal for sharing lessons in very specific contexts or on subjects where problems and solutions are constantly changing [12].

The upper left quadrant represents the Informal / Connect approach. In this case, discussion groups emerge from the common interest, allowing questions to be asked, answers to be given and lessons to be exchanged in a network of contacts without rigidity. Examples are the social networks found on the internet, such as LinkedIn and Facebook. The biggest difficulty is in ensuring that the answers come from reliable experiences [12].

The Formal / Collect approach (lower right quadrant) indicates an organized and managed system for the collection and retrieval of information, such as a database of lessons learned. They have the advantage of being able to track, locate, classify and group lessons and new knowledge. It is used in organizations such as the armed forces and industrial companies, where the consequences of not learning can lead to loss of life and also to a large financial loss [12].

2.2 Information Retrieval Systems

A textual information system is an application that helps people to access and analyze relevant information quickly and accurately, and that can consist of several modules, as illustrated in Figure 3. Initially, a content analysis module based on natural language processing techniques is required. This module allows the information system to transform raw text into more meaningful representations, which can be used more effectively by a search engine or any text analysis system. From this content treatment, several applications can be carried out, both in the scope of information retrieval (for access to information) and in the scope of data mining (for knowledge acquisition) [13].

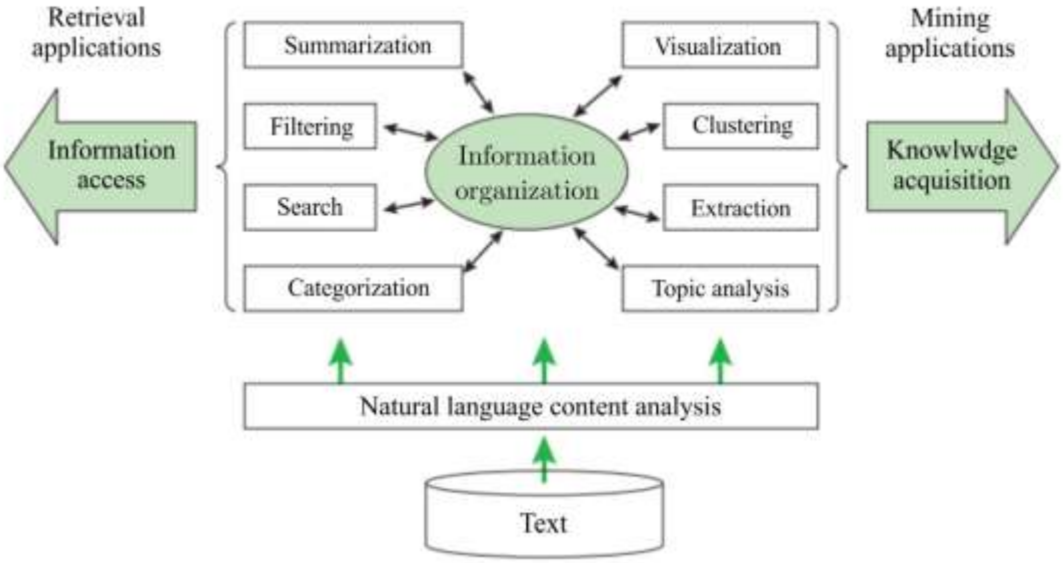


Figure 3. Conceptual framework for a text information system [13].

According to [14], information retrieval is the task of finding relevant documents and applicable to users' needs. The best examples of use are internet search engines, such as Google, whose operating algorithm is called PageRank [15]. An information retrieval system is characterized by four components:

- Documents: also called corpus, it is the data set to be analyzed.
- Query: also called a query, it is the information provided by the user about what needs to be searched in the set of documents.
- Results: subset of the documents that the information retrieval system identified as relevant to the incoming query.
- Presentation: it is the visual display of the results to the user.

3 Materials and Methods

The elaboration of this work followed the steps: data collection, extraction of texts and tabulation in the database, application of tokenization pre-processing techniques, removal of stop words and stemming of documents, preparation of a query for searching and ordering in the respective documents using the Okapi BM25 technique and elaboration of an interface to display the recovered documents.

3.1 Data collection

The collection of data is the initial phase, in which it is carried out the survey of text data to be used. Datasets in general contain a sequence of texts in documents $\mathbf{D = \{D_1, D_2, ..., D_n\}}$ where $\mathbf{D_i}$ refers to each document [16]. 1,500 public industrial safety alert documents in English obtained in PDF (Portable Document Format) from the following institutions were used: Center for Chemical Process Safety (CCPS), United States Coast Guard, Bureau of Safety and Environmental Enforcement (BSEE), National Offshore Petroleum Safety and Environmental Management Authority (NOPSEMA), International Association of Drilling Contractors (IADC) and National Agency of Petroleum, Natural Gas and Biofuels (ANP).

3.2 Extraction of texts and tabulation in database

From the collected documents, the extraction of the text contents was performed as well as its tabulation in a database. The FTS5 (Full Text Search) module [17] was used, which is a virtual table in the SQLite database that allows full-text searches to be performed in the databases. This type of mechanism allows you to search a large set of documents.

3.3 Pre-processing

Parallel to the stage of text extraction and tabulation in the database, a pre-processing was performed on the texts using the Python programming language and the NLTK (Natural Language Toolkit) library [18]. This step refers to an initial preparation to enable searches on the tool. These steps included tokenization, stop words removal and stemming [16].

Tokenization is the task of dividing the text (words or phrases) into parts called tokens. Stop words removal is a type of filter in which the words that appear frequently in the text are removed, but in general, do not have much information about content (for example, prepositions and conjunctions). Stemming is technique in which words are reduced to their motto.

In the developed tool, the steps of tokenization, removal of stop words and stemming were applied.

3.4 Search and sort using Okapi BM25

Most information retrieval systems are based on bag of words statistics. A ranking function receives a document and a query and returns a numerical indicator, with the most relevant documents receiving indicators with higher values. Among the ranking functions, there is the BM25, which was developed in the Okapi project at London's City College [19] and has been used by search engines, such as the open-source project Lucene [20]. In the BM25 function, the indicator is a linear combination of the indicators for each word that make up the query and takes into account three factors: the first is the frequency with which a word appears in a given document (TF or Term Frequency); the second is the inverted frequency of documents (IDF or Inverse Document Frequency), which represents the relative count of documents that contain a certain word; and the third is the total length of each document [14].

In the implementation of the proposed tool, the function BM25, presented in Equation 1 was used, which returns a real value indicating the similarity between the results of documents **D** and the input query **Q**. To calculate this value, the function divides the query into components, and the better the similarity, the lower the value returned by the function.

$$BM25(D,Q) = \sum_{i=1}^{Npalavras} IDF(q_i).\frac{f(q_i,D).(k+1)}{f(q_i,D)+k.(1-b+b.\frac{|D|}{mediadl})} \quad (1)$$

In this Equation 1, **Npalavras** is the number of words in the query, **|D|** is the number of words in the document, and **mediadl** is the average number of words in all documents in the FTS5 table. The **k** and **b** values are constants set at 1.2 and 0.75 respectively.

The term **IDF(q_i)** represents the inverted document frequency of the query word **i**. It is calculated according to Equation 2, where **N** is the total number of documents in the table and **n(q_i)** is the number of lines containing at least one instance of the word **i**.

$$IDF(q_i) = log\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \quad (2)$$

The term **f (q_i, D)**, represented in Equation 3, is the frequency of occurrence of the word **i**, that is the number of occurrences of a certain word in a given record. However, to allow the possibility of entering different weights for different parts of the text records, the function implemented by [17] allows the definition of weights for the columns (title column, content column , etc.), where **w_c** is the weight associated with the column **c** and **n(q_i, c)** is the number of occurrences of the word **i** in the respective column.

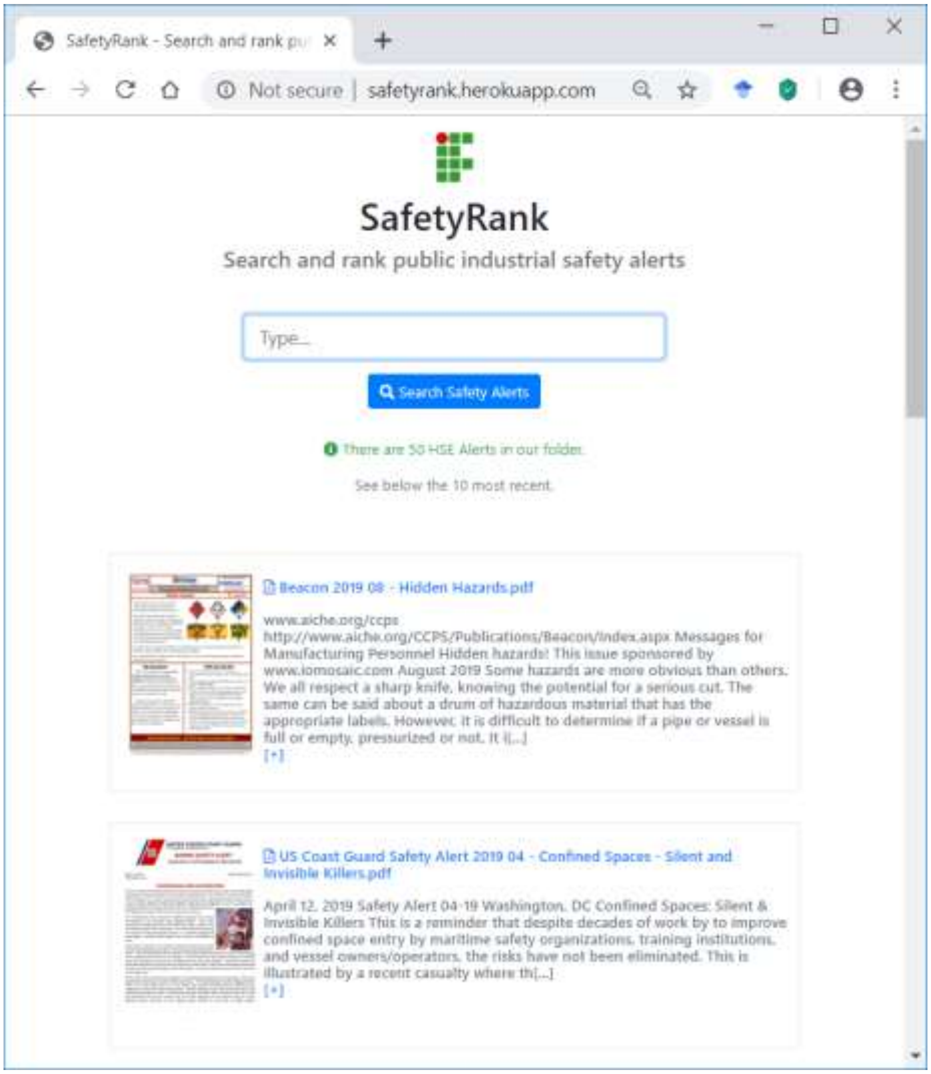
$$f(q_i,D) = \sum_{c=1}^{Ncoluna} w_c.n(q_i,c) \quad (3)$$

3.5 Development of an interface for displaying the results

For the elaboration of the tool interface the Flask framework was used, which is one of the most popular implementations of web application in the Python programming language [21]. This framework allows the construction of interfaces quickly and easily, as well as supporting the development of complex applications [22]. In addition to the web interface, an Application Programming Interface (API) functionality has been integrated into the tool, making it possible to carry out HTTP requests (Hypertext Transfer Protocol) by other applications, which information is exchanged in JSON (JavaScript Object Notation) format.

4 Results

The developed tool was made available in a free account on the Heroku platform (http://safetyrank.herokuapp.com/), which is a PaaS type cloud computing service (Platform as a Service) in which it is possible to manage web applications [23]. The source code was also made available (https://github.com/wanderfernandesjunior/safetyrank-information-retrieval). As shown in Figure 4, the tool was developed to receive only one entry from the user: a query with the words to be searched.


SafetyRank - Search and rank pu
Not secure | safetyrank.herokuapp.com
SafetyRank
Search and rank public industrial safety alerts
Type...
Search Safety Alerts

There are 50 HSE Alerts in our folder.

See below the 10 most recent.

Beacon 2019 08 - Hidden Hazards.pdf

www.aiche.org/ccps http://www.aiche.org/CCPS/Publications/Beacon/index.aspx Messages for Manufacturing Personnel Hidden hazards! This issue sponsored by www.iomosaic.com August 2019 Some hazards are more obvious than others. We all respect a sharp knife, knowing the potential for a serious cut. The same can be said about a drum of hazardous material that has the appropriate labels. However, it is difficult to determine if a pipe or vessel is full or empty, pressurized or not. It i[...]
[+]

US Coast Guard Safety Alert 2019 04 - Confined Spaces - Silent and Invisible Killers.pdf

April 12, 2019 Safety Alert 04-19 Washington, DC Confined Spaces: Silent & Invisible Killers This is a reminder that despite decades of work by to improve confined space entry by maritime safety organizations, training institutions, and vessel owners/operators, the risks have not been eliminated. This is illustrated by a recent casualty where th[...]
[+]

Figure 4. View of the developed search page.

The various industrial safety alert documents are dispersed across different websites of worldwide institutions, and the use of this tool has made it possible to centrally search, in a practical and quick way, for previous accident alerts that are relevant to a particular activity.

5 Conclusion

In this work, a computer program was developed to search for public industrial safety alerts related to accidents in the naval and oil and gas industry. The developed algorithm performs reading of the records, natural language processing (for tokenization, data cleaning and content stemming), indexing and storage in the database. In addition, the developed tool makes it possible to search for alerts, offering a ranking of the results by similarity with the input data using the Okapi BM25 method.

The Python programming language and the NLTK (Natural Language Toolkit) library were used to implement the text extraction, tabulation and pre-processing steps. The storage was performed in an SQLite database, in which a query ordered according to function BM25 was used. The results were displayed using a web page using the Flask framework.

Application Programming Interface (API) functionalities were integrated into the algorithm, making it possible to make requests by other applications. The use of this tool made it possible to search, in a practical and quick way, for previous accident alerts that are relevant to a certain activity.

As a future work, it is believed that it is necessary to increase the size of the database, either with public alerts or via data sharing agreements between companies, so that it is possible to further disseminate knowledge about accidents and safety alerts.

6 References

[1] SmartLab, Observatório de Segurança e Saúde no Trabalho; 2019. URL {https://smartlabbr.org/sst}, [Web; accessed in10-02-2020].

[2] Kletz, TA. Lessons from disaster: how organizations have no memory and accidents recur. IChemE; 1993.

[3] ANP, Alertas de Segurança - Agência Nacional do Petróleo, Gás Natural e Biocombustíveis; 2019.URL{http://www.anp.gov.br/exploracao-e-producao-de-oleo-e-gas/seguranca-operacional-e-meio-ambiente/resolucoes-notificacoes-procedimentos-e-orientacoes/alertas-de-seguranca}, [Web; accessed in18-05-2019].

[4] CCPS, Process Safety Beacon - Center for Chemical Process Safety; 2019.URL{https://www.aiche.org/ccps/resources/process-safety-beacon}, [Web; accessed in18-05-2019].

[5] USCG, Safety Alerts – United States Coast Guard; 2019. URL {https://www.dco.uscg.mil/Our-Organization/Assistant-Commandant-for-Prevention-Policy-CG-5P/Inspections-Compliance-CG-5PC-/Office-of-Investigations-Casualty-Analysis/Safety-Alerts/}, [Web; accessed in18-05-2019].

[6] IADC, Safety Alerts - International Association of Drilling Contractor;2019. URL {http://www.iadc.org/safety-alerts/}, [Web; accessed in 18-05-2019].

[7] Serrano, W. Neural Networks in Big Data and Web Search. Data 2019; 4(1):7.

[8] Kadhim, AI. Survey on supervised machine learning techniques for automatic text classification. Artificial Intelligence Review, 2019, p.1–20.

[9] Goodfellow, I; Bengio, Y; Courville, A. Deep learning. MIT press; 2016.

[10] Kulkarni, A; Shivananda, A. Natural Language Processing Recipes. Springer ;2019.

[11] Robertson, SE; Spärck Jones, K., Simple proven approaches to text retrieval. University of Cambridge, Computer Laboratory; 1994.

[12] Milton, N. The Lessons Learned Handbook: Practical approaches to learning from experience. Elsevier;2010.

[13] Zhai, C; Massung, S. Text data management and analysis: a practical introduction to information retrieval and text mining. Morgan&Claypool; 2016.

[14] Russell, SJ, Norvig, P. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited; 2016.

[15] Brin, S; Page, L. The anatomy of a large-scale hypertextual web search engine, 1998.

[16] Kowsari, K, Jafari Meimandi, K, Heidarysafa, M, Mendu, S, Barnes, L, Brown, D. Text Classification Algorithms: A Survey. Information 2019; 10(4):150.

[17] SQLITE, SQLite FTS5 Extension; 2019. URL {https://www.sqlite.org/fts5.html#_auxiliary_functions_}, [Web; accessed in 17-08-2019].

[18] Sphinx, Natural Language Toolkit; 2020. URL {https://www.nltk.org/}, [Web; accessed in 18-05-2020].

[19] Robertson SE, Walker S, Beaulieu M. Experimentation as a way of life: Okapi at TREC. Information processing & management 2000; 36(1):95–108.

[20] Apache, Class BM25 Similarity; 2017. URL {https://lucene.apache.org/core/7_0_1/core/org/apache/lucene/search/similarities/BM25Similarity.html}, [Web; accessed in 18-05-2020].

[21] FLASK, Flask; 2019. URL {https://palletsprojects.com/p/flask/}, [Web; accessed in 21-08-2019].

[22] Grinberg, M., Flask web development: developing web applications with python. O'ReillyMedia, Inc; 2018.

[23] HEROKU, What is Heroku? 2019. URL {https://www.heroku.com/about}, [Web; accessed in 17-08-2019].