Automation in Construction 62 (2016) 45–56

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports

Antoine J.-P. Tixier [a], Matthew R. Hallowell [a,*], Balaji Rajagopalan [b], Dean Bowman [c]

[a] Department of Civil, Environmental, and Architectural Engineering, University of Colorado at Boulder, Boulder, CO 80309, USA
[b] Department of Civil, Environmental, and Architectural Engineering, Cooperative Institute for Research in Environmental Sciences (CIRES), University of Colorado at Boulder, Boulder, CO 80309, USA
[c] Bentley Systems, Exton, PA, USA

ARTICLE INFO

Article history:
Received 12 March 2015
Received in revised form 20 October 2015
Accepted 7 November 2015
Available online 21 November 2015

Keywords:
Automated content analysis
Natural language processing
Text mining
Knowledge extraction
Accident
Injury
Safety
Attribute
Risk
R

ABSTRACT

In the United States like in many other countries throughout the globe, construction workers are more likely to be injured on the job than workers in any other industry. This poor safety performance is responsible for huge human and financial losses and has motivated extensive research. Unfortunately, safety improvement in construction has decelerated in the last decade and traditional safety programs have reached saturation. Yet major construction companies and federal agencies possess a wealth of empirical knowledge in the form of huge databases of digital construction injury reports. This knowledge could be used to better understand, predict, and prevent the occurrence of construction accidents. Unfortunately, due to the lack of a clear methodology and the high costs of manual large-scale content analysis, these valuable data have yet to be extracted and leveraged. Recently, researchers have proposed a framework allowing meaningful empirical data to be extracted from accident reports. However, the resource limitations inherent to manual content analysis still remain. The present study tested the proposition that manual content analysis of injury reports can be eliminated using natural language processing (NLP). This paper describes (1) the overall strategy and methodology used in developing the system, and specifically how key challenges with decoding unstructured reports were overcome; (2) how the system was built through an iterative process of coding and testing against manual content analysis results from a team of seven independent analysts; and (3) the implications and potential uses of the data extracted. The results indicate that the NLP system is capable of quickly and automatically scanning unstructured injury reports for 101 attributes and outcomes with over 95% accuracy. The main contribution of this research is to empower any organization to quickly obtain a large and highly reliable structured attribute and outcome data set from their databases of unstructured accident reports. Such structured data are a necessary prerequisite to the application of statistical modeling techniques, allowing the extraction of new safety knowledge and finally the amelioration of safety management.

1. Motivation

Construction is constantly ranked as one of the most dangerous industries worldwide [51]. In the United States, despite the improvements that followed the Occupational Health and Safety Act of 1970, construction still accounts for 17% of all work-related deaths while only employing 7% of the national workforce [17]. In fact, according to the Bureau of Labor Statistics [61], approximately 700 workers die each year. Construction fatalities and injuries result in immense societal costs, totaling approximately $15 billion in lost revenue every year [61]. What is even more alarming is that these colossal human and financial costs are expected to escalate with the 33% construction employment growth projections in the 2010–2020 decade, which is more than twice the overall anticipated economic growth [17].

Despite the abundant research that has been motivated by the aforementioned alarming injury and fatality rates, safety performance in construction has been plateauing in recent years, and the implementation of effective injury prevention practices has reached saturation [19]. Fortunately, risk-based approaches are emerging and show promise for safety improvement through proactive decision making. For example, Baradan and Usmen [6] compared the risk of building trades, Hallowell and Gambatese [62] quantified the safety risk for various activities required to construct concrete formwork, and Shapira and Lyachin [63] studied the impact of tower cranes on jobsite safety. However, these approaches are currently limited because (1) they

* Corresponding author.
E-mail addresses: antoine.tixier-1@colorado.edu (A.J.-P. Tixier), matthew.hallowell@colorado.edu (M.R. Hallowell), rajagopalan.balaji@colorado.edu (B. Rajagopalan), dean.bowman@bentley.com (D. Bowman).

http://dx.doi.org/10.1016/j.autcon.2015.11.001
0926-5805/

focus on specific activities and trades without considering the temporal and spatial interactions among risk factors, (2) they are not based on empirical data, and (3) they are limited in scope of application [48,51]. Consequently, existing models do not translate well to other work scenarios and do not capture the dynamics of construction work, where trades and activities constantly interact [31,51]. To overcome these limitations, Esmaeili and Hallowell [19] proposed a unified attribute-based framework that allows standard risk factor and outcome variables to be extracted from naturally occurring accident reports. Although this method shows promise, it requires the analysis of large numbers of reports to see patterns and trends emerge from the data. Such manual content analysis is laborious and resource-intensive [18,48].

To remove the needs for manual analysis of injury reports and allow the large-scale use of the attribute-based framework, the present study tests the proposition that attributes and safety outcomes can be automatically and accurately extracted from unstructured injury reports using natural language processing (NLP).

1.1. Background: attribute-based approach to construction safety

The attribute-based approach to construction safety theorizes that any construction situation can be uniquely and comprehensively characterized by a finite number of observable fundamental construction site attributes [18,48]. These basic elements are context-free, universal, and pertain to construction means and methods, environmental conditions, and human factors. For instance, in the following excerpt of an injury report, "employee was welding overhead and wind shifted, resulting in discomfort to left eye," three fundamental attributes can be identified: (1) welding, (2) working overhead, and (3) wind.

Although this approach is simple, it is very powerful. First, from this perspective, any incident can be viewed as the resulting outcome of the presence of a worker and the joint occurrence of some fundamental attributes. Consequently, attributes are also called *injury precursors* or simply *precursors*. In what follows, the terms *attribute* and *precursor* are used interchangeably. It is important to note that precursors should be observable before an injury occurs. *Falling object*, for example, is not a precursor, it is an outcome. On the other hand, *object at height* is a precursor. As illustrated in the previous example, descriptors of the work environment and outcomes can be extracted even from brief reports. Finally, this information is authentic since it is simply based on objective narratives of discrete events.

A connection with genetics can naturally be made with this style of analysis: every person is unique, but their genetic information is entirely encoded by combinations of a finite number of basic universal building blocks that constitute their DNA. The attribute-based approach to construction safety is built upon a similar theory that by identifying fundamental and universal construction injury precursors, understanding how they interact, and modeling how they shape risk and create unsafe work conditions, it may be possible to better understand the true nature of, predict, and prevent the occurrence of construction injuries. Historically, scientific understanding of complex phenomena has improved when breaking down convoluted systems into fundamental constituents that individually are easier to comprehend. A fascinating recent example is the Human Genome Project [16], which allowed sequencing and mapping of about 30,000 genes, unlocking the structure of human DNA. Similarly, the finite element method, a numerical technique used in many quantitative disciplines of engineering, is built on the theory that complicated continuous structures and objects can be represented by a finite number of geometrically simpler pieces [60].

Esmaeili and Hallowell [19] conducted the first attribute-level risk analysis in construction by analyzing 300 struck-by injury cases from national databases. Through this analysis, they identified 34 fundamental attributes. More recently, a team of eight researchers performed a manual content analysis of 2201 industrial injury reports gathered from 476 contractors, allowing the initial list of Esmaeili and Hallowell [19] to be refined and broadened to 80 precursors [18,48]. These precursors are summarized in Table 1. The validity of the content analysis and the relevance of the attributes presented by Prades [48] and Desvignes [18] were ensured by adhering to a strict coding scheme, implementing an iterative process with team-based calibration meetings, and using peer reviews and random checks by external reviewers. In these studies, attributes were classified in three categories: upstream, transitional, and downstream. Upstream precursors can be anticipated as soon as during the design phase, transitional precursors are generally not identifiable by designers but can be detected before construction begins based on knowledge of construction means and methods, and downstream precursors are mostly related to human behavior and can only be observed during the construction phase.

In addition to the 80 precursors presented in Table 1, Prades [48] and Desvignes [18] also extracted various safety outcomes from accident reports, namely, injury type, injury severity, body part affected, and energy source involved. The variables belonging to these categories are listed in Table 2. The injury codes, severity levels, and body divisions included in Table 2 are consistent with OSHA definitions and past research [30]. Energy sources were extracted based on the theory that any injury can be associated with the release of an energy source [23,28]. For instance, a falling load can be seen as a *gravity* release, a welding flash burn involves *radiation*, and waterproofing substances, solvents, or concrete in its liquid form can cause *chemical* burns. Additional definitions and examples can be found in Albert et al. [3]. Attributes and outcomes are occasionally referred to as variables in the rest of this paper.

Although the work of Esmaeili and Hallowell [19], Prades [48], and Desvignes [18] made important contributions to attribute-level safety analysis, the manual content analysis procedures used were time consuming, limiting the number of reports that could be analyzed in a reasonable research effort, and thereby the emergence of trends and patterns in the data extracted. For example, Desvignes [18] only used a random set of 1280 reports from a larger set of 4458 available reports because of time and resource limitations. In addition, even when a rigorous protocol is followed, it is never possible to entirely eliminate inconsistencies among human coders. For all these reasons, resorting

Table 1
Context-free validated injury precursors from Desvignes [18].

Upstream	Rebar	Screw
Cable tray	Scaffold	Slag
Cable	Soffit	Spark
Chipping	Spool	Slippery walking surface
Concrete liquid	Stairs	Small particle
Concrete	Steel sections	Adverse low temperatures
Conduit	Stripping	Unpowered tool
Confined workspace	Tank	Unstable support/surface
Congested workspace	Unpowered transporter	Wind
Crane	Valve	Wrench
Door	Welding	Lifting/pulling/manual handling
Dunnage	Wire	Light vehicle
Electricity	Working at height	Exiting/transitioning
Formwork	Working below elevated workspace/material	Sharp edge
Grinding	Drill	Splinter/sliver
Grout	**Transitional**	Repetitive motion
Guardrail/handrail	Bolt	Working overhead
Heat source	Cleaning	**Downstream**
Heavy material/tool	Forklift	Improper body position
Heavy vehicle	Hammer	Improper procedure/inattention
Job trailer	Hand size pieces	Improper security of materials
Lumber	Hazardous substance	Improper security of tools
Machinery	Hose	No/improper PPE
Manlift	Insect	Object on the floor
Stud	Ladder	Poor housekeeping
Object at height	Mud	Poor visibility
Piping	Nail	Uneven walking surface
Pontoon	Powered tool	

Table 2
Outcome categories from Prades [48] and Desvignes [18].

Injury code	Injury severity	Body part affected	Energy source involved
Caught in or compressed	Pain	Head	Biological
Exposure to harmful substance	First aid	Neck	Chemical
Fall on same level	Medical case	Trunk	Electricity
Fall to lower level	Lost work time	Upper extremities	Gravity
Overexertion	Permanent disablement	Lower extremities	Mechanical
Struck by or against	Fatality		Motion
Transportation accident			Pressure
			Radiation
			Thermal

to manual content analysis to systematically mine large databases of construction injury reports is not viable. In order to eliminate the needs for manual analysis of reports, and allow very large databases to be leveraged, this study introduces a fully automated and highly accurate NLP system.

In construction safety research, the only known attempt of automatically analyzing injury reports was made by Esmaeili [21]. In this study, high-severity injury reports contained in national databases were scanned for 22 attributes using commercial software. Although this effort involved extracting attributes from injury reports for the first time, it suffers some limitations. First, the reliability of the attribute identification and keyword validation process is questionable because fewer than 500 accident reports were used to identify attributes and tune keywords, and a percent agreement score of 0.7 was set as the accuracy threshold. This is a rather low value, especially since percent agreement is known to be a lenient metric that inflates agreement in all cases [35,43]. Second, the validated list of keywords and explanations about how NVivo was used to automate the content analysis were not provided, making replication of the work and assessment of the quality of the data obtained impossible. Third, only high-severity struck-by injuries were studied, which significantly narrowed the breadth of attributes that could be identified and the quantity, relevance, and generalizability of the keywords found. Fourth, some precursors were defined in opposition with the conceptual attribute-based framework. For instance, *falling object*, *structure collapse*, or *falling out from heavy equipment* cannot be considered injury precursors because they are outcomes rather than observable characteristics of the jobsite. This paper seeks to collectively address these limitations.

1.2. Background: natural language processing

Natural language processing (NLP) is a very active and rapidly evolving area of research that deals with the comprehension and analysis of human-produced texts by computers [15]. It enables machines to derive meaning from human language input. NLP lies at the confluence of artificial intelligence, linguistics, and computer science and aims at achieving human-like natural language understanding [42]. Applications of NLP include speech recognition, machine translation, and automated content analysis [44].

Automated content analysis is increasingly being used for a variety of applications. This can be explained by the needs to make sense of and leverage the fast-growing volume of digital information [5]. In construction, even a small project generates a lot of electronic information in the form of specifications, computer-aided drawings, process control, inventory management, cost estimating, scheduling, and other documentation [53]. For many years, major companies and federal agencies have also been constituting extensive databases of electronic injury and near miss reports as part of their safety programs.

In a recent study, Francis and Flynn [24] attempted to categorize insurance claim descriptions into four categories based on keywords and then used the clustering to predict claim severity. For instance, the claims corresponding to car accidents were automatically extracted based on the keywords *hit*, *travel*, and *vehicle*. In another study, 10,000 traffic incident reports were automatically categorized into topics using latent Dirichlet allocation and incorporated into predictive models to forecast time-to-clearance and improve traffic management in real time [46]. In the construction industry, text analytics have been used to classify project documents [2,12], to retrieve computer-aided drawings from databases [34], to automate knowledge extraction from narratives and represent it as a map [59], and to structure and manage safety knowledge in order to support job hazard analysis [14,57].

Creating and validating a system that understands natural language is very challenging. Natural languages are complex, consisting of a catalogue of words, called a lexicon, and set of structural rules, called a grammar, that allows meaning to be built by combining words into sentences [44]. Historically, the most serious obstacle to effectively analyze naturally occurring language has been the difficulty to accurately model grammars [32]. Additionally, many words have several meanings, making the use of word sense disambiguation indispensable.

Most modern natural language processing tools use machine-learning algorithms and statistical modeling to overcome the aforementioned barriers. Some of the most widely used techniques in text analytics are clustering algorithms, such as k-means, and classification algorithms, such as k-nearest neighbors, naïve Bayes, and support vector machines [27,39,41,44,47,55]. Some popular methods also include random forest [27], graph theory [64], Bayesian and Markov models [5], and latent Dirichlet allocation [46].

2. Selection of an appropriate natural language processing method

In developing the automated content analysis tool, a dilemma arose. Ideally, machine-learning algorithms would have been applied to the available data manually coded by Prades [48] and Desvignes [18]. Unfortunately, these techniques, such as support vector machines, perform poorly when a sufficient number of positive training examples are not available for each category [47]. Some researchers have suggested that to attain effective statistical learning, 75 to 100 positive cases per category is an absolute minimum [8,33]. Due to the high dimension of the injury report feature space (80 attributes) and the diversity of construction situations, the available training data were naturally sparse. For instance, in the report "carpenter felt discomfort in his left knee while exiting a tight area," only 2 attributes are present, namely, *exiting/transitioning* and *confined workspace*. The other 78 attributes are not featured. Similarly, only a couple of attributes co-occur in each injury report. Therefore, for a large number of reports, cases when a given attribute is present are outnumbered by cases when this same attribute is not present. For example, in Desvignes' [18] data set of 1280 manually analyzed reports, the median attribute in terms of number of appearances, *heavy vehicle*, occurs only 21 times. Therefore, manually analyzing tens of thousands of injury reports would have been required to put together a satisfactory training database and achieve efficient machine learning.

Because such a large number of reports was not available, and because of time and resource limitations, the team decided to design an NLP program based on hand-coded rules and dictionaries of keywords.

Although this approach is not as simple and elegant, it offers advantages over machine learning. Most importantly, it allows researchers to integrate human intelligence and knowledge of the data into the system, allowing higher levels of accuracy to be reached [52]. As Wang et al. [56] described, statistical classifiers are targeted at broad and relatively shallow understanding, whereas handcrafted rules perform well within a specific domain when deep understanding is sought. Changes in coding scheme, detection of new variables, and higher skill can also be achieved very quickly by simply updating the rules and dictionaries, whereas algorithms require new, expensive training data to get better. Additionally, rule-based tools avoid the relatively opaque nature of machine learning [7,10].

3. Design of a rule-based automated content analysis system

The R programming language [49] was used to develop a fully automated NLP system based on hand-coded rules and dictionaries of keywords. Although there are other languages that could have been used, R was preferred because it is open source, increasingly being used, and features high-quality libraries to more efficiently build on the previous work of others. The coding scheme, operational definitions of variables, and knowledge gained from the manual content analysis of more than 2200 injury reports [18,48] were used to design the tool. Examples of injury reports from Desvignes's [18] database are provided in Table 3. Although these reports are not lengthy, they were written by personnel working on site within 8 h of the injury as requested by the participating contractors' policies and contain enough details to have a good idea of the work environment at the time of the accident and the injury outcomes. Other report examples can be found in Table 5.

The system was built to automatically and accurately scan injury reports for the 80 validated attributes, 7 injury types, 5 body parts, and 9 energy sources summarized in Tables 1 and 2. Some minor differences with Desvignes's [18] original classification are to be noted: *object at height on same story* was grouped with the attribute *object at height*, and the precursors *slag* and *sparks* were separated. The attribute *snow/ice* was extended to include low temperature incidents (e.g., cases of hypothermia) and was renamed *adverse low temperatures*. The attributes *unpowered hand tool* and *powered hand tool* were extended to include tools that are not handheld but do not belong either to the category *machinery* (e.g., girder spacer, power trowel, etc.). Therefore, these attributes were renamed *powered tool* and *unpowered tool*. Also, one attribute, *improper procedure/inattention*, was added. Finally, the injury types *struck by* and *struck against* were grouped under the umbrella *struck by or against*, which is consistent with the Occupational Injury and Illness Classification System (OIICS) and BLS classifications [30].

As shown in Fig. 1 and as previously mentioned, NLP systems generally consist of a lexicon, or catalogue of words, and a set of structural rules, called a grammar, allowing meaning to be derived from the way words are combined [44]. Fig. 2 describes the tool building and validation process. In what follows, details about the construction of the lexicon and grammar are given.

3.1. Step 1: lexicon building

The first major step in the design of the automated content analysis system was the development of keyword dictionaries. Although there has been a great deal of construction safety research, no lexicon related to precursors of construction injuries was available at the time of this research. To create such a lexicon, several resources were leveraged as shown in Fig. 2 and as discussed below.

The first resource for this inquiry was the 2201 manually analyzed incident reports from Prades [48] and Desvignes [18]. These data include the attributes and outcomes shown in Tables 1 and 2 for each report, thereby allowing systematic sorting and identification of common keywords and phrases linked to each attribute. Online resources, such as the OSHA website, were subsequently used for dictionary enrichment. The focus of this section of the paper is on this decomposition process and the creation of the lexicon.

3.1.1. Specific keywords

In order to initiate keyword dictionary creation, the R "tm" package [36] was used. The hand-analyzed reports were sorted by attributes and outcomes and the most frequent terms associated with each group of reports were found. For example, some of the most frequent words associated with the energy source *chemical* were "burn," "irritation," "line," "liquid," "concrete," "water," "chemical," "caustic," "eye," "acid," "cloud," "face," "coker," "laborer," "burning," "skin," "insulation," "insulator," "mist," "splatter," "waterproofing," "sprayed," and "flushed." Although only a simple frequency count, this list is very insightful. A first observation can be made that some of the keywords in the list, such as "caustic," "acid," and "chemical," are very specific to the energy source *chemical*. Indeed, the sole occurrence of these terms in a given incident report would suffice to classify the report as a *chemical* injury. Single keywords are known as unigrams [44], and "caustic," "acid," and "chemical" can thus be considered to be unigrams specific to the energy source *chemical*.

Within the context of *chemical* energy, the keyword "concrete" is an excellent example where additional words were required to derive meaning and to properly identify when concrete-related incidents were related to the chemical properties of the material. For instance, "concrete pouring" denotes a task, "concrete bag" a heavy material, "concrete drill" a powered tool, "concrete blanket" a type of tarpaulin, "concrete foreman" a person, and "concrete burn" refers to a type of chemical burn. Accordingly, although the sole presence of "concrete" or "burn" does not allow the variable *chemical* to be detected (i.e., "concrete" and "burn" are not unigrams specific to *chemical*),

Table 3
Examples of injury reports, with attributes and outcomes.

Reports	Attributes	Energy source	Injury code	Body part
"A finisher apprentice was applying crystalline waterproofing to construction joints and cracks inside a pontoon cell. At some point the waterproofing material got in-between his kneepad and wet jeans which caused a concrete burn on his leg. The area was treated with a burn spray and he employee returned back to work immediately."	Hazardous substance, pontoon	Chemical	Exposure to harmful substance	Lower extremities
"As employee was walking, he stepped on a nail that was lying in the road base/gravel. The nail was lying free on the ground. Employee felt the nail puncture his foot and immediately pulled it out and reported it to the safety representative who was nearby."	Nail, uneven walking surface, object on the floor	Motion	Struck by or against	Lower extremities
"Employee was lifting a 2 × 12 wood plank when the wood plank got too heavy causing it to fall back towards the employee and hit him on the top/front of his hard hat."	Lumber, heavy material, lifting/pulling, improper security of materials	Gravity	Struck by or against	Head
"At the ABC site wet tailrace a worker went to the Sea-Can for tools and PPE. When he went to open the door of the Sea-Can he received a 120 volt shock."	Door, electricity	Electricity	Exposure to harmful substance	Not detectable
"A pipefitter was welding on a pipe support when slag fell into the cuff of his left glove, resulting in a burn to his left wrist."	Welding, steel sections, slag	Thermal	Exposure to harmful substance	Upper extremities

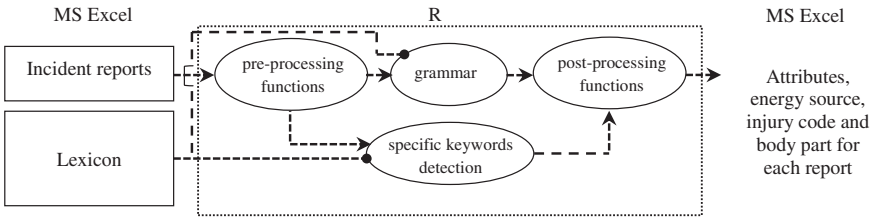

Fig. 1. Overarching NLP tool process flow.

when these two words are found in a report as a pair (i.e., "concrete burn"), there is no ambiguity that the report deals with a *chemical* incident. Thus, "concrete burn" can be considered to be a specific double keyword (i.e., a specific bigram), associated with the energy source *chemical*.

The most frequent terms for all variables were carefully inspected, and unigrams, bigrams, and trigrams specific to each of the 101 variables (i.e., 80 attributes, 9 energy sources, 7 injury types, and 5 body parts) were gathered. These terms were then arranged in a lexicon composed of one catalogue of terms, or dictionary, for each variable. Online resources allowed the dictionaries to be enriched in order to anticipate new cases. For instance, synonyms of "caustic," such as "corrosive," "irritative," "toxic," etc., can be looked up and added to the dictionary for *chemical* even if these particular keywords are not found directly in the reports. Other online resources such as the OSHA website also contain a tremendous amount of valuable keywords and were used to augment the dictionaries even more. Most of the keywords found in the final dictionaries were gathered by adopting this "generalization and anticipation" process as shown in Fig. 2. Some examples of these dictionaries of specific keywords include *steel sections* (37 unigrams, 80 bigrams, 2 trigrams), *lumber* (46 unigrams, 34 bigrams, 26 trigrams), and *poor housekeeping* (4 unigrams, 3 bigrams). Although specific keywords allow easy variable detection in some cases, they are not sufficient for detecting all variables in all cases, and it is necessary to look at generic keyword combinations to decipher meaning from text.

3.1.2. Generic keywords

In many cases, variables have to be detected based on combinations of keywords that are not specific to any variable. These keywords are referred to as generic keywords. Looking closely at the list of keywords for the energy source *chemical*, one can note that some keywords, such as "insulator," "laborer," "eye," or "skin," are related to persons, while some others refer to actions ("flushed," "sprayed"), materials ("insulation"), outcomes ("irritation"), or location (a "coker" is an oil refinery unit). None of these keywords alone, if found in an injury report, would guarantee the presence of the variable *chemical*, nor of any other variable. However, *chemical* should be detected if "insulation" is associated with "eye," "skin," "irritation," or "burning." Therefore, all the keywords related to the idea of irritation (e.g., "irritation," "itching," "burning," etc.), all the keywords dealing with the human body (e.g., "skin," "eye," "hand," etc.), and all the keywords about insulation (e.g., "insulation," "fiberglass," "glass wool," "foam," etc.) were collected and stored in dedicated dictionaries (adopting the anticipation and generalization process previously discussed), and a grammatical rule for the detection of the energy source *chemical* was created. Another example when an attribute can be identified based on generic keywords is *working at height*. This precursor should be detected when a term linked to the topic of working (e.g., "working," "doing," "performing," "drilling," "installing," etc.) is associated with a keyword involving the notion of height (e.g., "height," "elevation," "elevated," "mezzanine," "roof," etc.). These simple examples are provided to illustrate the spirit of the variable detection strategy, but in practice, relying simply on association is not enough to avoid false alarms. Detection rules (also called grammatical rules) have to take into account the order in which the keywords appear and the context. The rule building process will be described in detail in the next section.

Since generic keywords do not allow specific variables to be detected until they are combined with other keywords, generic keywords were not organized in variable-related dictionaries but rather stored in topic or concept-related dictionaries. Some of these 47 dictionaries are, for

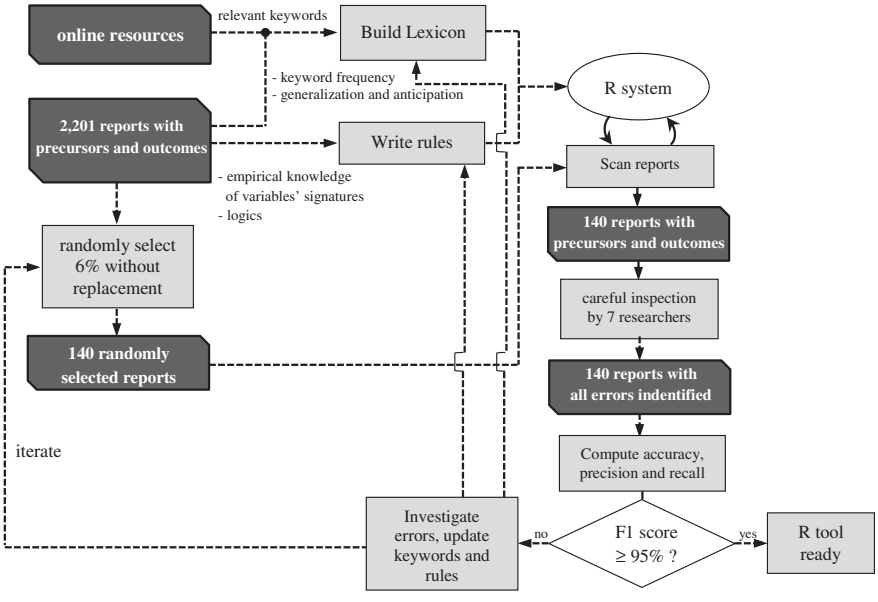

Fig. 2. NLP tool building and validation process.

example, *people* (141 unigrams, 8 bigrams, plus the 475 most given first names in the U.S.), *working* (117 unigrams, 1 bigram), *elevation* (21 unigrams, 1 bigram, 1 trigrams), and *unstable* (47 unigrams, 24 bigrams).

The notions of specific and generic words originate from the field of linguistics. For example, Ferrer I Cancho and Sole [64] observed, using graph theory, that there are two different regimes of words: basic and specialized. Also, it is important to keep in mind that dictionary creation is an iterative process. No lexicon is fully comprehensive, and careful inspection of the system's errors, deep understanding of the textual data, and full use of available linguistics resources is required to ensure a continuous improvement of the dictionaries [27]. Such updating, while time consuming and sometimes tedious, is necessary [40]. The most recent version of the lexicon that was developed for this research is available on request by emailing the corresponding author. As shown in Fig. 2, and as will be discussed, errors were closely examined during the validation process, and keywords and rules were tuned accordingly.

3.2. Step 2: devising detection rules

The second step in the development of the system was the writing of grammatical rules to detect variables based on combinations of generic keywords in the text. Detecting variables using generic keywords is complex. Simply testing for co-occurrence of topics creates many errors. To illustrate, consider the following two injury report excerpts:

Excerpt 1: "moving floor plate, employee strained his back."

Excerpt 2: "employee rolled his ankle on moving floor plate."

In each excerpt, the generic keywords in bold are associated with the same topics and appear in the exact same order. Yet, very different precursors should be identified in each case. In the first report, it should be inferred that a back injury was sustained as a result of the presence of the attribute *material handling*, whereas in the second report, the worker sprained their ankle due to the occurrence of the precursor *unstable support/surface*. The ambiguity comes from the fact that the bigram "floor plate" has two different senses: (1) material and (2) walking surface. If "floor plate" refers to some material, then it is being moved (necessarily by a person), and "moving" is a verb. On the other hand, if "floor plate" designates a walking surface, then it is moving as a result of an unintended action or some exterior influence and "moving" is used as an adjective to qualify "floor plate."

To detect the correct attribute, it is necessary to determine which of the senses of "floor plate" is invoked in each report, or in other words, to disambiguate the sense of the generic bigram "floor plate." In many instances, disambiguation is helped by looking at the context, such as looking at preceding and following words. Also, taking into account prepositions such as "on," "onto," "into," "under," and "over" gives a lot of information. Finally, the structure of the injury reports can be decomposed by using punctuation marks such as commas and periods and conjunctions such as "and," "or," and "while." In the given example, it should be noted that the attribute *unstable support/surface* can be rightfully identified in the second report if the fact that the preposition "on" precedes "floor plate" is captured.

Twenty-six variables were detected based on their specific keywords only. For the energy source *biological*, for instance, the occurrence of "biosludge," "scorpion," "bees," "bugs," or "wastewater" (and others very specific, unambiguous keywords) was enough to trigger detection. However, for 76 of the 101 variables, it was necessary to account for the subtleties previously mentioned, namely, (1) neighboring generic or specific keywords (context), (2) report structure (punctuation, conjunctions), and (3) ordering of terms. As shown in Fig. 2, the signature of each of these variables was empirically determined from knowledge gained during the manual content analysis of more than 2200 reports, from generalization, anticipation, and grammatical logic, and from lessons learned during the tool tuning process. These signatures were transposed into the R programming language in the form of hand-rafted detection rules.

Within the 76 variables requiring grammatical rules, 51 were detected based on a combination of specific keywords and rules, and 25 did not have specific keywords and were detected based on rules alone. An example of an attribute with mixed rules is *confined workspace*, where "potholing," "manhole," "tunnel," and others were used as specific keywords. To capture all other cases, the variable was detected if any element from the generic family *confined* (e.g., "confined," "limited," "tight," "narrow," etc.) was found combined with any element from the generic family *area* (e.g., "area," "room," "space," "quarters," "entrance," etc.).

The remaining 25 variables were detected based on rules alone. For instance, the grammatical rule for the attribute *object at height* can be written in English as "any element from the topic *materials.tools* is followed by any element from the topic *fall*) AND (no element from the topic *people* is found sandwiched between any element from the topic *materials.tools* and any element from the topic *fall*) AND (any element from the topic *elevation* is present)." For *cable*, the rule is much simpler and can be written in English as any of the keywords ("cable," "cables") is present, but is not immediately followed by any of the keywords ("tray," "shovel," "wheel," "reel," "spool," "coil," or "trench"). Indeed, *cable tray* is another attribute, a cable shovel is classified as a *heavy vehicle*, wheels, reels, coils, and spools are classified in the attribute *spool*, and "cable trench" is not specific to *cable* (e.g., "carpenter tripped on cable trench"). Note that this is done without loss of generality since both "cable" and "cable trench" are free to co-occur. In the sentence "worker was installing cable in cable trench," *cable* would be detected.

To efficiently write these statements in the R programming language, a library of custom functions was developed.

3.2.1. Creation of R functions

Table 4 shows the comprehensive list of functions that were created to write the rules. Every single rule was written as a combination of statements involving these functions. Before being passed to these functions, as shown in Fig. 1, unstructured text needs to be cleaned and converted into structured data. Functions from the "base" [49] and "tm" [36] R packages were used to perform these preprocessing steps.

Preprocessing included (1) converting the text to lower case (R is case sensitive), (2) removing some punctuation marks, (3) eliminating words containing little or no information (known as stop words), and finally (4) stripping the extra white space. In removing punctuation (step 2), commas and periods were kept since they provide valuable information about text structure. For example, in "accident involved one welder. Falling hammer from mezz deck struck him," the immediate proximity between the two keywords "welder" and "falling" should be disregarded because these two words do not belong to the same sentence. Hence, the machine should understand that the welder himself is not falling. The third step, stop words removal, is standard and used to some degree in all text-mining applications (e.g., [12,46]). Stop words such as "what," "too," "a," "be" and others were removed, but words referring to persons, such as "she," "he," "they," "his," "her," etc., as well as other words like some prepositions and conjunctions (e.g., "below," "between," "into," "and," "so," etc.), proved useful and were kept. Numbers and intra-word dashes were also kept since a number of keywords include such characters. For instance, some specific unigrams for the attribute *lumber* are "2 × 2," "2 × 6," and some for the precursor *bolt* are "she-bolt" and "u-bolt." Finally, the last step consisted in splitting the text based on whitespace. This action turns unstructured text into an ordered character vector. The elements of the vector can be words, punctuation marks (commas and periods), and numbers. Each element is automatically assigned an index number, which corresponds to the position of the element in the ordered vector of words (i.e., in the text). This step is fundamental, as it allows distance between elements, called "radius" in what follows, to be measured.

Table 4
Comprehensive list of functions used for writing rules.

Function	Returns true if… (false else)	ORIGIN
Statement 1 | statement 2	(Statement 1 is true) or (statement 2 is true)	R base package
Statement 1 and statement 2	(Statement 1 is true) and (statement 2 is true)	
Any (a % in % *x*)	Any unigram of the character vector[a] a is present in the character vector x	
Any(sapply(a, grepl, text))	Any unigram, bigram, or trigram of the character vector a[b] is present in the text	
Complex.s(a, b, radius, text)	Any element of the character vector a is followed by any element of the character vector b in the text, in the same part of the sentence, and within the radius provided.	Developed in R for this research
Tricky.double(a, b, c, number, radius, text)	• If number = 1 any element of the character vector a is present in the text, but is not followed by any element of the character vector b within the radius provided, • If number = 2 any element of the character vector b is present in the text, but is not preceded by any element of the character vector a within the radius provided, • If number = 3 any element of the character vector b is present in the text but is not preceded by any element of the character vector a within the radius provided, nor followed by any element of the character vector c within the radius provided	
Sandwich.wrap(a, b, c, text)	Any element of the character vector c is not sandwiched between any element of the character vector a and any element of the character vector b	

[a] The term "character vector" simply refers to an ordered vector of elements. The character vectors *a*, *b*, and *c* are used to represent the content of specific and generic keywords dictionaries. The order in which the elements appear in the dictionaries does not matter. On the other hand, the character vector *x* is the text of the injury report split based on whitespace (last preprocessing step previously described), so the order matters and corresponds to the order in which elements appear in the text.
[b] In this case, the elements of *ss* have to be regular expressions.

The key assumption when working with radii is that only words close to each other are related. Beyond a certain distance, words are not connected anymore. This is consistent with the Markov assumption in natural language processing, which holds the local context of a word only to be of importance [44]. In time series analysis, an equivalent approach is used: the correlation between observations decreases and eventually disappears as temporal distance increases ([58], pp.6-26).

The functions summarized in Table 4 offer the same kind of flexibility than the NEAR, AFTER, BEFORE, OR, AND, and NOT functions proposed in the commercial content analysis software Wordstat 6.0 [65]. In addition, rules are limited to a maximum of 2 function-based statements in Wordstat. Of course, R-based rules are not limited in their number of statements, so variables can be detected based on more intricate patterns, and finer tuning can be achieved. Also, in Wordstat, the items passed to each function can only be picked from a limited number of pre-defined categories. The R functions developed for this research can be used for any combination of elements (unigrams, bigrams, trigrams, punctuation marks, and numbers). These combinations can take the form of multiple dictionaries assembled, subsets of dictionaries, and keywords punctually prescribed. There is no limitation in quantity and origin of the elements. These functions are available on request by emailing the corresponding author.

3.2.2. Illustrative example

An example of rule building using some of the R functions previously introduced will be presented. These examples consider the incident report: "the employee was walking down the stairs and slipped." After being cleaned and structured into an ordered vector of elements, this report can be represented as shown in Fig. 3.

If the goal is to detect the attribute *slippery walking surface*, one rule can be written in English as "any element corresponding to the idea of *people* is followed by any element from the topic *slipping* within a radius of 7." This rule can be translated in R as a single complex.s() statement and is capable of detecting the variable *slippery walking surface*, as illustrated in Fig. 4.

However, caution should be used when working with radii. If the distance prescribed is too short, variables can go undetected (a radius of 3 in the example at hand, for example). However, if the distance prescribed is too long, the risk of capturing spurious relationships and thus of improperly detecting variables is increased, as shown in Fig. 5.

In order to minimize the risk of such false alarms, the rule can be modified as follows: "(any element corresponding to the idea of *people* is followed by any element from the topic *slipping* within a radius of 7) AND NOT (any element referring to the topic *slipping* is preceded by any element of the topic *tools* within a radius of 3)." This rule can be written in R using two complex.s() statements, as shown in Fig. 6. With such a new rule, the false alarm is avoided.

Furthermore, because only *persons* and (*materials* or *tools*) can slip, the rule can be simplified as "any element from the topic *slipping* is present, but this element is not preceded by any element from the topic *materials.tools* within a radius of 3." This is written in R as a single tricky.double() statement. Some other noteworthy examples of when tricky.double() statements prove useful include "line of fire" (*thermal* should be detected when "fire" is present but not when "fire" is immediately preceded by "line of"), "chain fall," (a chain fall is a tool and has nothing to do with *falling*), "concrete vibrator" (*concrete* should be detected when "concrete" is present but not when it is immediately followed by "vibrator," "hammer," etc.), "rebar foreman" (*rebar* should be detected when "rebar" is present but not when it is immediately followed by "foreman," "finisher," etc.), "chipping hammer" (*chipping* should be detected when "chipping" is present, but not when it is immediately followed by "hammer"), or "finger nail," "nail gun," etc. (*nail* should be detected when "nail" is present, but not when it is immediately preceded by "finger," "thumb," "index," etc., nor followed by "gun," "driver," etc.).

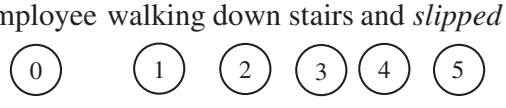

Fig. 3. Cleaned and structured incident report.

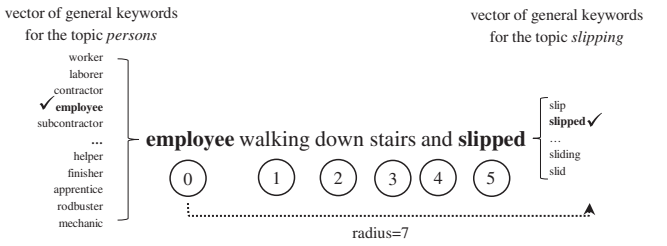

Fig. 4. Detection of the variable *slippery walking surface* based on a single complex.s() statement.

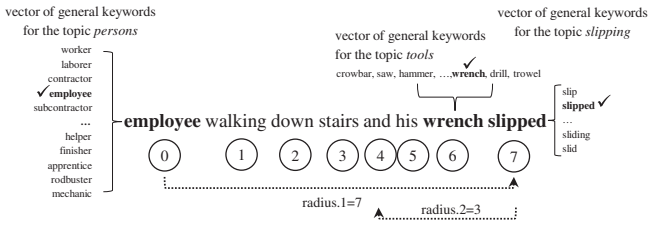

Fig. 6. Two complex.s() statements allow the false alarm to be avoided.

3.3. Processing incident reports

When presented with reports, as shown in Fig. 1, the NLP tool starts by selecting the first report. The report is cleaned and scanned for any specific keyword associated with any variable. Then detection rules for all variables are tested. When all attempts to detect variables have been made, a binary vector of length the total number of variables is returned. The binary vector features "1" whenever the corresponding variables have been detected, and "0" elsewhere. These steps are repeated for all the other reports. Because all reports are scanned in sequence, parallel processing could be used to speed up the process. For that purpose, the "doParallel" R package [50] was used. By using 36 cores at 2.6 GHz, 4377 reports could be analyzed in just under 11 min. After the tool is done looping through all the reports, a binary matrix is obtained, as illustrated by Fig. 7. The following step consists in resolving conflicts.

3.4. Resolving conflicts among detected variables

The reasoning behind using a clash detection and resolution function is that some variables are incompatible, such as *working at height* and *working below elevated workspace*, while some others are implicitly connected, like *radiation* and *exposure to harmful substance*. After a given report has been scanned, it is thus necessary to make sure that the variables that have been detected are not incompatible and that no natural association has been missed.

For example, if the keyword "ice" is present, the attribute *ice* will be detected (except in cases where ice packs are applied on bruises, or ice buckets are carried, etc.). However, because topics about *persons* and *slipping* will always be present in ice-related incident reports, and moreover combined in the same fashion ("employee slipped on […]"), the attribute *slippery walking surface* will always be detected if *ice* is detected. While not fundamentally incorrect (*ice* is indeed a subset of *slippery walking surface*), this association is problematic. In fact, the attribute-based framework strives to produce high-quality structured data on which statistical models can be applied. For that purpose, overlaps in the attributes are to be avoided as much as possible to ensure that every precursor keeps its full predictive power. Therefore, conflict resolution rules were created. In this example, the rule is to delete *slippery walking surface* whenever *ice* is present. Examples of similar conflict resolution rules include (but are not limited to) *sparks* and *small particles* (*sparks* is preferred), *stairs* and *ladder* (*ladder* is preferred), and *fall on same level* and *fall to lower level* (*fall to lower level* is preferred).

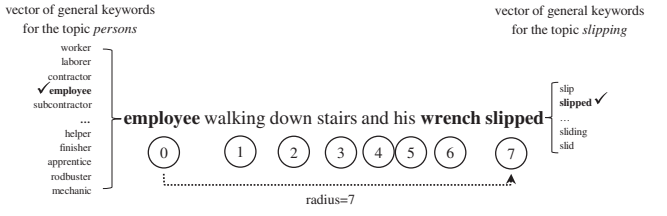

Fig. 5. Faulty detection of the variable *slippery walking surface* by a single complex.s() statement.

In addition to rules precluding redundancies, other rules were developed to ensure proper association. Indeed, some variables should always be found together. For example, the injury code *struck by or against* should always be associated with the energy source *motion*. In the same manner, the presence of the energy sources *electricity*, *thermal*, or *chemical*, should always trigger the detection of the injury code *exposure to harmful substance*. Finally, if the injury codes *fall to lower level* or *fall to same level* have been detected, the energy source *gravity* should always be added. Such rules serve as an internal control in the coding structure and were integrated into the conflict.resolution() function.

As described in Fig. 7, when all conflicts have been resolved by the conflict.resolution() function, the binary matrix is converted to a textual matrix by the fill.names() function. This matrix comprises injury reports as rows and the attributes and outcomes detected as columns (see Table 5). Both matrices (binary and textual) are automatically written as Excel spreadsheets and can also be analyzed directly in R.

4. Validation of the system and measurement of reliability

As shown in Fig. 2, the tool tuning process was iterative. At each step, 140 injury reports were randomly drawn without replacement from the data set of 2201 reports of Desvignes [18] and Prades [48]. These reports were then automatically scanned by the R system, and as previously explained, a textual matrix of 140 rows by 19 columns (1 column for the textual reports, 15 for the attributes, 1 for energy source, 1 for injury code, and 1 for body part) similar to Table 5 was returned. This output was divided into seven textual matrices of 20 rows each. Each table was assigned to a researcher, who had been involved with the Desvignes [18] study and was familiar with the coding scheme and operational definitions of variables. As illustrated by the retrieval matrix in Table 6, each researcher reviewed their randomly assigned piece of output looking for true positives (TP), false positives (FP), and false negatives (FN).

True positives (TP), also informally called "hits," refer to cases when the tool has rightfully detected a precursor that was indeed present in the injury report. False positives (FP), also called "false alarms," or type I error, designate an instance when the tool has wrongfully detected an attribute that was not present in the injury report (not relevant). Finally, false negatives (FN), also called "misses," or type II error, are cases when the tool has omitted to detect the presence of a precursor that was actually there in the report (not detected and relevant). It should be noted that the last option, true negatives (TN), occurs when the tool has not detected an attribute that was not present in the injury report. True negatives were not taken into account.

After careful examination by the seven researchers, the reviewed textual matrices were aggregated and all true positives, false positives, and false negatives were counted. Three performance metrics, standard in the field of information retrieval and NLP, were then computed: precision, recall, and F-1 score [2,34,11]. It should be noted that inspection of the system's output agreed upon by seven humans was preferred over automatic comparison to a gold standard (e.g., manually labeled reports). Indeed, this allowed the source of each error made by the

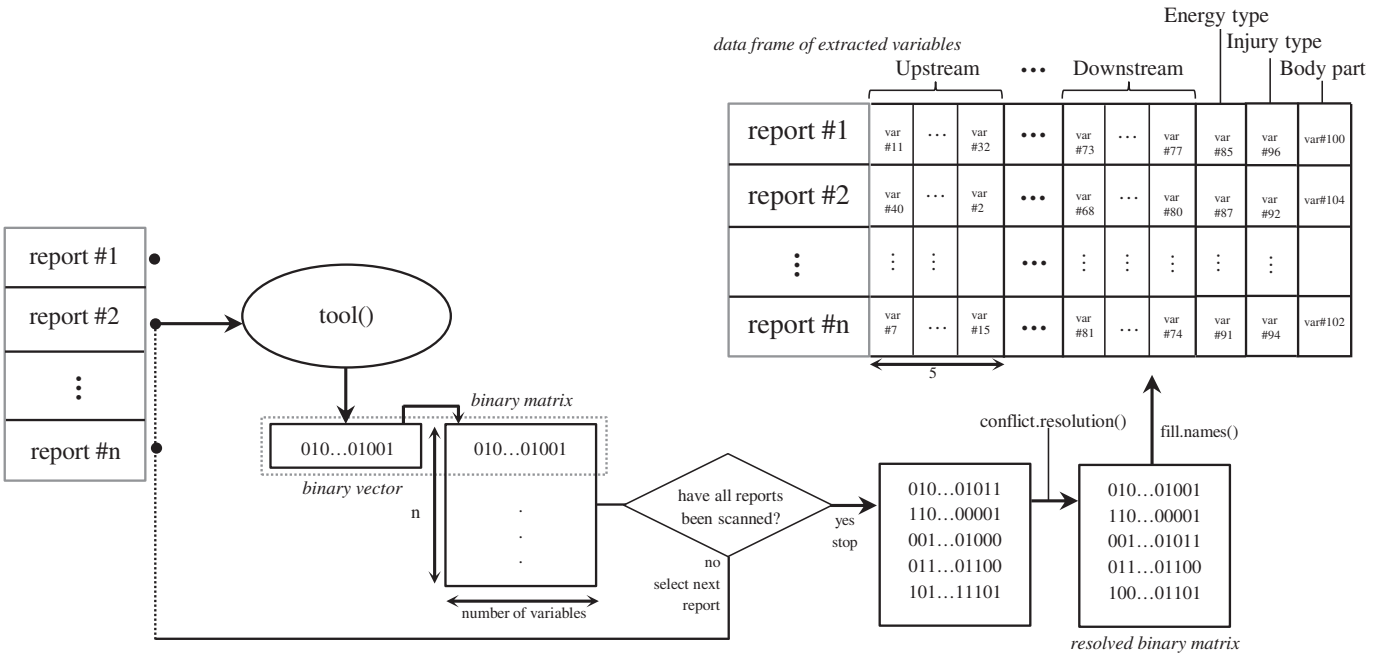

Fig. 7. Scanning of injury reports, and post-processing.

system to be investigated, identified, and fixed by tuning the grammatical rules and lexicon accordingly. This output examination process played a crucial role in developing a highly accurate system.

Precision is calculated, as shown in Eq. (1), as the proportion of relevant items to the number of items detected [11]. This is simply the probability that an attribute is present given that it was detected by the tool [26]. High precision means that most results returned are relevant. Maximum precision is attained in the absence of type I error (i.e., no false positive). On the other hand, recall is the number of retrieved relevant items as a proportion of all relevant items (see Eq. (2)). In other words, recall is the probability that a precursor that is present is detected by the tool. A high recall rate means that most of the relevant results are returned, and recall is maximized in the absence of misses (i.e., in the absence of type II error). Buckland and Gey [11] define precision as the purity of retrieval and recall as the completeness of retrieval:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{1}$$

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{2}$$

More precisely, the formulas that were used macro-averaged the results [47]. Instead of considering recall and precision rates for each variable separately (what is known as micro-averaging), true positives, false positives, and false negatives for all categories were aggregated, and the recall and precision rates averaged these counts. Macro-averaging treats each class equally and is harsher than micro-averaging since one class that results in a bad performance can significantly deteriorate the overall performance [47].

Finally, the F-1 score is defined as the harmonic mean of precision and recall. As shown in Eq. (3), the F-1 score gives the same weight to precision and recall, assuming that the cost of a false positive equals the benefit of a true positive.

$$\text{F-1 score} = 2\,\frac{\text{precision}\;\text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

A threshold of 95% in F-1 score was selected as a tool tuning stop criterion. This is a high threshold, especially when using strict macro-averaging performance metrics, but as noted by Grimmer and Stewart [27], it is crucial to show that the automated program is able to replicate human coding. The scores for each round of random reviews are summarized in Table 7. Four iterations were needed before the 95% threshold in F-1 score was achieved. As shown in Fig. 2, lessons learned from careful examination of the errors made by the system played a huge role in improving skill.

The error rates for energy source and injury code, simply defined as the number of errors divided by the number of reports scanned ($n = 140$), were also computed at each round. For body part, to avoid any false alarm, the tool was designed to return *not detectable* when more than one body part is detected, or when the information is not present in the report. The detection of body part was only added to the system's functionality after the third iteration of random reviews, and *not detectable* was returned 6.25% of the time. When any body part was detected, the system was correct every time.

These scores are comparable or even better than the scores attained by most statistical classifiers found in the literature. For instance, Verma et al. [55] used a naïve Bayes classifier to analyze tweets and reached 80% accuracy. Tweets were to be classified into 5 categories. A final accuracy of 0.65 with a recall rate of 0.75 was obtained by Grimmer and Stewart [27], with a random forest classifier. Go et al. [25] used unigrams and bigrams as variables to classify tweets as positive or negatives and an accuracy of 83% was reached. Finally, Bai [5] reviewed two studies about automated opinion mining and movie review polarity categorization and reported that the accuracies ranged between 66% and 88.9%.

In the construction field specifically, Al Qady and Kandil [2] implemented unsupervised clustering algorithms to classify project documents into mutually exclusive classes and reached a 0.844 *F*-score in the optimum case. The computer-aided drawings automated retrieval system of Yu and Hsu [34] attained 100% recall, but the precision was only 57.2%. Finally, 86.37% accuracy was reached by Caldas and Soibelman [12], who used support vector machines to classify construction project documents into hierarchical classes.

It is not surprising that the system based on handcrafted rules developed in this research outperformed all the aforementioned

Table 5
Example of the system's output for 12 injury reports.

Description	up.1	up.2	up.3	trans.1	trans.2	trans.3	down.1	energy	code	body part
Employee was welding on a pipe, as he brought hands down he touched the tungsten with left finger resulting in a burn.	Piping	Welding						Thermal	Exposure to harmful substance	Upper extremities
The employee was in the process of hoisting a piece of cable tray to an above level and he scraped his arm on a sharp edge of the cable tray.	Cable tray			Unpowered tool	Lifting pulling manual handling	Sharp edge		Motion	Struck by or against	Upper extremities
Climbing out of scaffold and felt back discomfort.	Scaffold	Working at height		Exiting				Motion	Overexertion	Trunk
Employee was grinding a pipe in a tight spot, grinder kicked back making contact with right thumb resulting in an abrasion.	Confined workspace	Grinding	Piping	Powered tool				Mechanical	Struck by or against	Upper extremities
EE was lifting a 2 X 12 wood plank when the wood plank got too heavy causing it to fall back towards the EE and hit him on the top/front of his hard hat.	Heavy material/tool	Lumber		Lifting pulling manual handling			Improper security of materials	Gravity	Struck by or against	Head
Welding FOB	Welding			Small particle				Motion	Struck by or against	Head
Employee was offloading burners with a cart when the cart moved unexpectedly "crushing" his leg between the cart and the existing railing.	Unpowered transporter	Guardrail/handrail		Lifting pulling manual handling				Motion	Caught in/compressed	Lower extremities
Employee was grinding overhead, weight shifted and board he was standing on slid, causing a "pop" to upper leg resulting in a strain.	Grinding	Lumber			Working overhead	Unstable support/surface		Motion	Overexertion	Lower extremities
Laborer suffered concrete burns during a chipping operation.	Chipping							Chemical	Exposure to harmful substance	Not detectable
The employee reported that he received an insect bite/sting on 6/21/13 at work.				Insect				Biological	Exposure to harmful substance	Not detectable
Employee was walking out of the warehouse building. The floor was wet from the rain. He slipped and fell on his left knee on the concrete floor.	Concrete	Slippery walking surface	Exiting					Gravity	Struck by or against	Lower extremities
An employee of … Mechanical was using oxy/acetylene to cut a pipe when hot slag entered his sleeve and burned his wrist which was treated on site. Cutting gloves were not being used.	Piping	Welding		Slag			No or improper PPE	Thermal	Exposure to harmful substance	Upper extremities

Table 6
Retrieval matrix (adapted from Buckland and Gey [11]).

	Relevant	Not relevant
Retrieved	TP	FP
Not retrieved	FN	TN

machine-learning-based tools since as noted by Sagae and Lavie [52], hand-coded rules allow researchers to transfer their expertise, knowledge of the data, and human intelligence to the system. Tools based on manually devised rules are usually capable of deeper understanding than machine-learning classifiers when the domain of application is very specific [56].

5. Conclusion, limitations, and recommendations

Major construction firms and federal agencies have been recording injury-related events and near misses in the form of digital textual reports for many years, but due to the lack of an adapted framework and methodology, and due to the high time, labor, and organizational costs related to manual content analysis (not to mention inter-coder reliability issues), the most part of this valuable knowledge is left unstructured and unexploited. Specifically, low-severity, high-frequency events that are not OSHA-recordable but account for huge financial and long-term human costs are very seldom studied.

This research tested the proposition that the needs for manual content analysis of incident reports can be eliminated using NLP. Results clearly show that this is possible. The R system that was developed in this research is capable of scanning naturally occurring, unstructured textual injury reports for 101 relevant, valid and carefully defined variables (80 precursors, 7 injury types, 9 energy sources, and 5 body parts) with high recall (0.97) and precision (0.95) rates. The keyword dictionaries and all R functions developed are available on request by emailing the corresponding author. Some functions are also provided in the appendix. As will be discussed in the recommendations section, the proposed NLP system will enable organizations to quickly and automatically extract the knowledge contained in their databases of unstructured injury reports to improve safety management.

The attribute-based framework of Esmaeili and Hallowell [19], refined by Prades [48] and Desvignes [18], showed great promise with respect to extracting useful and standardized information from unstructured injury reports. However, the large-scale use of this framework was previously limited by the high costs and issues related to manual content analysis. Consequently, all the subsequent analyses necessary to identify trends and patterns in the attribute and outcome data (i.e., extract insight) were limited by the small amount of data available. The tool developed in this study removes the need for manual content analysis of injury reports, enabling very high numbers of reports to be scanned for attributes and outcomes, statistical modeling to be applied on very large data sets, and more insight to be extracted.

Table 7
System's performance at each step of the tuning process.

Iteration	Precision	Recall	F-1 score	Error rate for energy source	Error rate for injury code	"Not detectable" rate for body part
1	0.854	0.924	0.888	11.5%	8.0%	NA
2	0.911	0.946	0.928	8.6%	4.3%	NA
3	0.917	0.959	0.938	3.6%	2.9%	NA
4	0.950	0.970	0.960	5.7%	5.7%	6.25%

5.1. Limitations

First, the system is inherently limited by the use of hard detection rules: it is not robust to unfamiliar and erroneous input, such as misspelled, missing, and unseen words. In other words, the system cannot address situations that were not anticipated, and the quality of the available textual data directly affects the quality of the attribute and outcome data extracted by the system. For instance, when faced with a misspelled word (e.g., "steal" instead of "steel") the R system is unable to detect the associated variable *steel/steel sections*. Most frequent misspellings can be anticipated, but obviously, it is unfeasible to account for all possible cases. Also, some reports contain a description of the events following the incident (e.g., "[…] worker was brought to the job trailer and an ointment was applied"). Despite the many precautions that were taken, precursors can still be wrongfully detected from these irrelevant portions of the text.

The R system reached very high scores during random reviews of its output, indicating that the impact of erroneous or misleading input was very limited. Indeed, the reports available to this research were generally carefully written and only contained facts relevant to the incident. However, verifying the quality of the injury reports is a necessary first step that should always be taken before running the R tool on any new database in the future. Reports of overly poor quality may prevent the automated content analysis from being successfully conducted.

The methodology introduced in this study is applicable to other domains. Especially, the system structure (i.e., the overall approach, methodology, and rule-writing functions that were developed) is ready to be used in any situation. Also, the full system is ready to scan incident reports pertaining to the industrial, energy, infrastructure, and mining fields as it is, since the grammatical and conflict resolution rules as well as the lexicon developed in this research were tuned and validated on such reports. If scanning reports belonging to other industry sectors are wanted, a prerequisite for reaching high skill will be to extend the tool's dictionaries and detection rules. Defining new fundamental attributes may also be needed, which requires trained content analysts and calibration meetings. As Grimmer and Stewart [27] note, a limitation of dictionary-based methods is that they are only efficient inside the domain for which the dictionaries were originally developed.

Finally, using the tool requires installing R on the machine. Knowledge of R is not needed, as the process is fully automatic (a single command only has to be executed by the user). However, tuning the tool (i.e., creating/updating rules) requires basic R literacy.

5.2. Recommendations for future research

When a sufficient amount of reports have been classified by the R system in each category, a natural next step would consist in hybridizing the system with machine-learning algorithms such as support vector machines. As a hybrid, the system would keep the deep understanding given by hand-coded rules while acquiring the flexibility of statistical classifiers. Moreover, by judiciously aggregating the decisions of hard and soft detection rules, the skill of the system could increase even more [47].

In order to get closer to 100% accuracy, the errors made by the tool can be automatically detected using data-mining methods such as hierarchical clustering, which is known for its ability to isolate outliers in small clusters. Manually inspecting these small clusters allows easy identification and correction of the errors. This approach can be used for instance to attain a maximal signal over noise ratio before training statistical predictive models on the data.

The main contribution of the proposed NLP system is its ability to extract meaningful structured attribute and outcome data from unstructured injury reports automatically and with high accuracy. Such structured data represent raw material required to apply data-mining and statistical modeling algorithms that will allow to better understand,

predict, and prevent the occurrence of construction accidents. For instance, models predicting safety outcomes from combinations of attributes will assist safety managers in accurately diagnosing the safety risk associated with specific construction situations and provide them with tailored recommendations based on simple observations of the work environment. The proposed system can be seen as a construction safety knowledge discovery tool: large databases of injury reports represent a wealth of valuable lessons waiting to be learned and made readily available to construction professionals in order to help decision making and to prevent mistakes from being repeated over and over.

The methodology presented in this research can be applied to mine other construction textual data like contracts and project documentation, but more generally, it can be applied to mine any kind of text. The use of NLP may soon become mandatory and widespread in construction management to make sense of the ever-growing amount of digital information associated today with even the smallest construction project. Statistical classifiers and machine-learning algorithms are without a doubt the present and the future of text analytics, but this study shows that when very high accuracy is sought on a specific, well-defined domain, dictionaries and hand-coded rules can bring better results, once a one-time investment has been made.

Acknowledgments

We would like to thank the National Science Foundation for supporting this research through an Early Career Award (CAREER) Program. This material is based upon work supported by the National Science Foundation under grant no. 1253179. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would also like to recognize Bentley Systems for their financial support for this research and intellectual contributions.

References

[2] M. Al Qady, A. Kandil, Automatic clustering of construction project documents based on textual similarity, Autom. Constr. 42 (2014) 36–49.
[3] A. Albert, M.R. Hallowell, B. Kleiner, A. Chen, M. Golparvar-Fard, Enhancing construction hazard recognition with high-fidelity augmented virtuality, J. Constr. Eng. Manag. 140 (7) (2014).
[5] X. Bai, Predicting consumer sentiments from online text, Decis. Support. Syst. 50 (4) (2011) 732–742.
[6] S. Baradan, M.A. Usmen, Comparative injury and fatality risk analysis of building trades, J. Constr. Eng. Manag. 132 (5) (2006) 533–539.
[7] D. Barbella, S. Benzaid, J.M. Christensen, B. Jackson, X.V. Qin, D.R. Musicant, Understanding Support Vector Machine Classifications via a Recommender System-Like Approach, *DMIN* July 2009, pp. 305–311.
[8] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, J. Popp, Sample size planning for classification models, Anal. Chim. Acta 760 (2013) 25–33.
[10] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[11] M.K. Buckland, F.C. Gey, The relationship between recall and precision, J. Am. Soc. Inf. Sci. 45 (1) (1994) 12–19.
[12] C.H. Caldas, L. Soibelman, Automating hierarchical document classification for construction management information systems, Autom. Constr. 12 (4) (2003) 395–406.
[14] N.W. Chi, K.Y. Lin, S.H. Hsieh, Using ontology-based text classification to assist job hazard analysis, Adv. Eng. Inform. 28 (4) (2014) 381–394.
[15] G.G. Chowdhury, Natural language processing, Annu. Rev. Inform. Sci. Technol. 37 (1) (2003) 51–89.
[16] F.S. Collins, E.D. Green, A.E. Guttmacher, M.S. Guyer, A vision for the future of genomics research, Nature 422 (6934) (2003) 835–847.
[17] CPWR, The Center for Construction Research and Training, produced with support from the National Institute for Occupational Safety and Health grant number OH009762, 2013.
[18] M. Desvignes, Requisite empirical risk data for integration of safety with advanced technologies and intelligent systemsMaster's Thesis University of Colorado, Boulder, 2014http://dx.doi.org/10.1061/9780784412329.030.
[19] B. Esmaeili, M. Hallowell, Attribute-Based Risk Model for Measuring Safety Risk of Struck-By Accidents, Construction Research Congress 2012 2012, pp. 289–298, http://dx.doi.org/10.1061/9780784412329.030.
[21] Behzad Esmaeili, Identifying and quantifying construction safety risks at the attribute levelPhD Dissertation University of Colorado, Boulder, 2012.
[23] M.A. Fleming, Hazard recognition, *By Design*, ASSE, 2009 11–15.
[24] L. Francis, M. Flynn, Text mining handbook, *Casualty Actuarial Society E-Forum, Spring 2010*, Vol. 1, 2010.
[25] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, CS224N Project Report, Stanford 2009, pp. 1–12.
[26] C. Goutte, E. Gaussier, A probabilistic interpretation of precision, recall and F-score, with implication for evaluation, Advances in Information Retrieval, Springer, Berlin Heidelberg 2005, pp. 345–359.
[27] J. Grimmer, B.M. Stewart, Text as data: the promise and pitfalls of automatic content analysis methods for political texts, Polit. Anal. (2013) (mps028).
[28] W. Haddon, Energy damage and the ten countermeasure strategies, Hum. Factors 15 (4) (1973) 355–366.
[30] M.R. Hallowell, A formal model for construction safety and health risk management, ProQuest, 2008.
[31] M.G. Helander, Safety hazards and motivation for safe work in the construction industry, Int. J. Ind. Ergon. 8 (3) (1991) 205–223.
[32] D. Hindle, Acquiring disambiguation rules from text, Proceedings of the 27th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics June 1989, pp. 118–125.
[33] D.J. Hopkins, G. King, A method of automated nonparametric content analysis for social science, Am. J. Polit. Sci. 54 (1) (2010) 229–247.
[34] J.Y. Hsu, Content-based text mining technique for retrieval of CAD documents, Autom. Constr. 31 (2013) 65–74.
[35] Dawn Iacobucci (Ed.) Journal of Consumer Psychology's Special issue on Methodological and Statistical Concerns of the Experimental Behavioral Researcher, 10 (1&2), Lawrence Erlbaum Associates, Mahwah, NJ 2001, pp. 71–73.
[36] Ingo Feinerer, Kurt Hornik, David Meyer, Text mining infrastructure in R, J. Stat. Softw. 25 (5) (2008) 1–54 (URL: http://www.jstatsoft.org/v25/i05/).
[39] A. Karatzoglou, I. Feinerer, Kernel-based machine learning for fast text mining in R, Comput. Stat. Data Anal. 54 (2) (2010) 290–297.
[40] W.L. Kuechler, Business applications of unstructured text, Commun. ACM 50 (10) (2007) 86–93.
[41] N. Li, D.D. Wu, Using text mining and sentiment analysis for online forums hotspot detection and forecast, Decis. Support. Syst. 48 (2) (2010) 354–368.
[42] E.D. Liddy, Natural Language Processing, 2001.
[43] M. Lombard, J. Snyder-Duch, C.C. Bracken, Content analysis in mass communication: assessment and reporting of intercoder reliability, Hum. Commun. Res. 28 (4) (2002) 587–604.
[44] C.D. Manning, Foundations of Statistical Natural Language Processing, in: H. Schütze (Ed.) MIT press, 1999.
[46] F.C. Pereira, F. Rodrigues, M. Ben-Akiva, Text analysis in incident duration prediction, Transp. Res. C Emerg. Technol. 37 (2013) 177–192.
[47] R. Prabowo, M. Thelwall, Sentiment analysis: a combined approach, J. Informetr. 3 (2) (2009) 143–157.
[48] M. Prades, Attribute-Based Risk Model for Assessing Risk to Industrial Construction tasksMaster's thesis University of Colorado, Boulder, 2014.
[49] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2014 (URL http://www.R-project.org/).
[50] Revolution Analytics, Steve Weston, doParallel: Foreach parallel adaptor for the parallel packageR package version 1.0.8 http://CRAN.R-project.org/package=doParallel2014.
[51] R. Sacks, O. Rozenfeld, Y. Rosenfeld, Spatial and temporal exposure to safety hazards in construction, J. Constr. Eng. Manag. 135 (8) (2009) 726–736.
[52] K. Sagae, A. Lavie, Combining Rule-Based and Data-Driven Techniques for Grammatical Relation Extraction in Spoken Language, *Proceedings of the Eighth International Workshop in Parsing Technologies*, Nancy, France, 2003.
[53] L. Soibelman, J. Wu, C. Caldas, I. Brilakis, K.Y. Lin, Management and analysis of unstructured construction data types, Adv. Eng. Inform. 22 (1) (2008) 15–27.
[55] S. Verma, S. Vieweg, W.J. Corvey, L. Palen, J.H. Martin, M. Palmer, ... K.M. Anderson, Natural Language Processing to the Rescue? Extracting "Situational Awareness" Tweets During Mass Emergency, ICWSM, July 2011.
[56] Y.Y. Wang, A. Acero, C. Chelba, B.J. Frey, L. Wong, Combination of statistical and rule-based approaches for spoken language understanding, *INTERSPEECH*, September 2002.
[57] H.H. Wang, F. Boukamp, Ontology-based representation and reasoning framework for supporting job hazard analysis, J. Comput. Civ. Eng. 25 (6) (2011) 442–456.
[58] W.W.S. Wei, Time Series Analysis, Addison-Wesley publ, 1994.
[59] C.L. Yeung, C.F. Cheung, W.M. Wang, E. Tsui, A knowledge extraction and representation system for narrative analysis in the construction industry, Expert Syst. Appl. 41 (13) (2014) 5710–5722.
[60] O.C. Zienkiewics, The Finite Element Method in Engineering Science, McGraw-Hill, Londres, 1971.
[61] Bureau of Labor Statistics, Census of Fatal Occupational Injuries (2014) http://www.bls.gov/news.release/cfoi.nr0.htm.
[62] M.R. Hallowell, J.A. Gambatese, Activity-based safety risk quantification for concrete form work construction, J. Constr. Eng. Manag. 135 (2009) 990–998.
[63] A. Shapira, B. Lyachin, Identification and analysis of facttors affecting safety on construction sites with tower cranes, J. Constr. Eng. Manage. 1351 (2009) 24–33.
[64] R.F. i Cancho, R.V. Solé, The small world of human language, Proc. R. Soc. Lond. B Biol. Sci. 268.1482 (2001) 2261–2265.
[65] Provalis Research, Content analysis and text mining software (2014) http://provalisresearch.com/products/content-analysis-software/.