Natural Language Processing for Safer Radiation Therapy

Abstract

Short

In healthcare, unintended incidents and accidents may recur if they are not reported and investigated as part of an incident learning program. As incidents are typically reported in free text, they can be difficult to investigate in large numbers. Here, we report on a project to facilitate incident learning in radiation therapy with natural language processing techniques that are applicable to incident learning across healthcare. First, we used supervised learning algorithms to help expert users label incidents by ordering suggested labels in a dropdown menu by their likelihood, estimated by multi-output regression. We assembled over 200 multi-output ensemble regressors by bagging or boosting over 60 regressors from scikit-learn, XGBoost, and Keras. We tuned the ensemble regressors with a grid-search, and cross-validated and compared them by how close to the top they placed expert-generated labels in the dropdown menu. As a result, a bagged support vector regressor, passive aggressive regressor, random forest regressor, and a long short-term memory recurrent neural network placed the expert-generated labels on average 1.57th, 2.36th, 5.25th, and 1.11th among the 9 process step, 24 problem type, 21 contributing factors, and 4 overall severity labels, respectively. Second, in a process known as topic modelling, we used an unsupervised learning algorithm to cluster the 3162 incidents into 44 topics, and the 634, 1275, and 333 incidents labelled as "other" for process step, problem type, and contributing factors into 24, 30, and 27 topics, respectively. We then explored the topics in a shareable interactive visualization which showed which incidents recurred the most, and how they were related to other incidents. This enabled us to comprehend large number of incidents without reading each, and therefore consider patient-reported incidents as well in incident learning of the future. In conclusion, natural language processing can facilitate incident learning in radiotherapy.

Long

Background

In healthcare, unintended incidents and accidents may recur if they are not reported and investigated as part of an incident learning program. As incidents are typically reported in free text, they are often difficult to investigate in large numbers. Here, we aim to facilitate incident learning in radiation therapy with natural language processing techniques that are applicable to incident learning across healthcare. First, we aim to use supervised learning algorithms to help label incidents by their process step, problem type, contributing factors, and overall severity. Second, we aim to use an unsupervised learning algorithm to cluster incidents into topics, and explored them in a shareable interactive visualization.

Methods

We preprocessed 3162 incident reports with a pipeline that included machine translation, abbreviation resolution, and biomedical named entity normalization, then explored term frequency–inverse document frequency and word embedding for text vectorization. In using supervised learning algorithms to help label an incident, we ordered the labels in a dropdown menu by their likelihoods, estimated by multi-output regression. We assembled over 200 multi-output ensemble regressors by bagging or boosting over 60 regressors from scikit-learn, XGBoost, and Keras. We tuned the ensemble regressors with grid-search, and cross-validated and compared them by how close to the top they placed the expert-generated labels in the dropdown menu. In using an unsupervised learning algorithm to cluster similar incident reports into topics, a technique known as topic modelling, we tuned a latent Dirichlet allocation model with multi-metric grid-search over the number of topics to best fit the incident reports, and interpreted the topics as possible problem types or contributing factors in an interactive visualization.

Results

A bagged support vector regressor, passive aggressive regressor, random forest regressor, and a long short-term memory recurrent neural network placed the expert-generated labels on

average 1.57th, 2.36th, 5.25th, and 1.11th among the 9 process step, 24 problem type, 21 contributing factors, and 4 overall severity labels, respectively. Then, the latent Dirichelet allocation model clustered the 3162 incidents into 44 topics, and the 634, 1275, and 333 incidents labelled as "other" for process step, problem type, and contributing factors into 24, 30, and 27 topics, respectively. The interactive visualization of the topics showed which incidents recur the most, and how they relate to other incidents.

Conclusion

Supervised learning algorithms can help an expert user to label incidents, thereby revealing trends in incident occurrence. Unsupervised learning algorithms can enable an expert user to comprehend a large number of incident reports without reading each individually, and therefore consider patient-reported incidents as well in incident learning of the future. In conclusion, natural language processing can facilitate incident learning and may enable a patient-centred approach to quality assurance in healthcare.

Introduction

Radiotherapy is a multi-step and highly complex treatment modality used in approximately two thirds of cancer treatments (Kaur 2011, Montgomery 2016). The complexity of the modality arises from the wide range of cancers treated, the technologies used, and the level of expertise required to prescribe, plan and deliver the radiation treatments. Although errors occur rarely, approximately once per six hundred treatments, the consequences to the patient can be severe and irreversible (Ford 2012, Mans 2010).

Incident learning, the process of identifying, reporting, and investigating patient safety incidents, and installing protocols that prevent their recurrence has been widely implemented and standardized in radiotherapy (Cunningham 2010, Huq 2016, Milosevic 2016). In radiotherapy specifically, we must learn from near misses as well rather than exclusively from actual incidents at great cost to patients. Learning from both actual incidents and near misses is widely practiced in industrial environments such as aviation and nuclear power plants, where

incidents have severe consequences. We can imagine the benefits of this learning in radiotherapy based on the benefits incident learning brought to these industries. As shown in Figure 1, it is estimated that for every fatal incident, there are thirty minor incidents and three hundred near misses (Jehring 1951). Recursive self-improvement by investigating minor incidents and near misses can be highly beneficial. For example, incident learning has reduced the frequency of aviation incidents by 73% in 10 years (Ford 2012).

Figure 1: An illustration of the relative occurrence of major accidents, minor injuries, and near misses in industrial environments.

Collaborative incident learning initiatives in radiotherapy have sprung up within institutions as well as within and between national and international communities. In Canada, the *National System for Incident Reporting–Radiation Treatment* (NSIR-RT) (Milosevic 2016) is implemented as a national online reporting and analysis system. The NSIR-RT taxonomy comprises 33 data elements pertaining to six categories: impact, discovery, patient, incident details, treatment delivery, and investigation. The taxonomy has generated a high level of agreement within the Canadian radiotherapy community, and is roughly aligned with the US,

European and International Atomic Energy Agency (IAEA) radiation treatment incident classification systems (Misosevic 2016).

In January 2016, the Department of Radiation Oncology of Cedars Cancer Centre of the McGill University Health Centre (MUHC) in Montreal, Canada deployed a software known as the *Safety and Incident Learning System* (SaILS) software for incident reporting and learning that includes interfaces for incident reporting, investigation, tracking, and data visualization (Montgomery 2016). The MUHC version of SaILS is compatible with the NSIR-RT taxonomy, and had processed 716 incidents between January 2016 and September 2018. Reporting an incident requires mainly a textual narrative of the incident and a one-sentence descriptor. For each incident that is reported, one or more investigators retrospectively and manually classify the incident in SaILS using menu choices corresponding to the data elements of the NSIR-RT taxonomy. For example, the process step of incident occurrence is one of the data elements of the NSIR-RT taxonomy, and it has the menu choices: patient assessment and consultation; imaging for radiotherapy planning; treatment planning; pre-treatment review and verification; treatment delivery; on-treatment quality management; and other. The reporting interface of SaILS is shown in Figure 2.

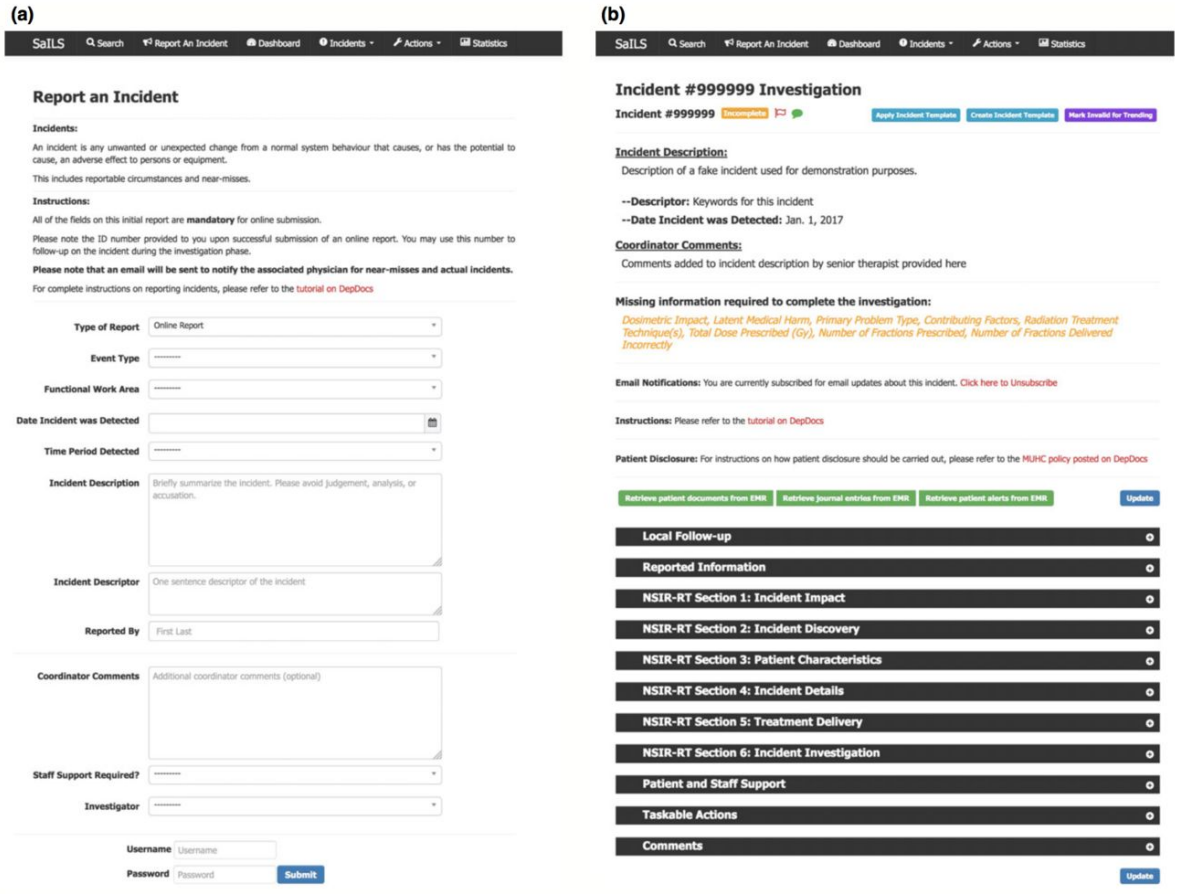


Figure 2: Screenshot of the reporting interface of the NSIR-RT-compatible version of the Safety and Incident Learning System (SaILS).

Healthcare incident reports, including the radiotherapy incidents submitted into the SaILS software, are similar to the safety reports of various industrial environments in that their narratives are reported in an unstructured free-text format. Free text, while convenient for the reporter, presents a challenge for data aggregation and requires suitably-qualified personnel to read and analyze. Alas, due to the lack of dedicated incident-analysis personnel, minor incident reports in healthcare often accumulate as resources are used to deal with front-line issues that are typically considered more urgent.

Natural language processing (NLP) is an emerging technique for analyzing large quantities of clinical texts, notably in high-technology medical specialties such as radiation oncology and

radiology (Meystre 2018, Pons 2016) . According to Meystre and Pons, the five major categories of application of NLP in radiation oncology are (1) diagnostic surveillance, (2) cohort building for epidemiological studies, (3) query-based case retrieval, (4) quality assessment of radiologic practice, and (5) clinical support services. In the present article, we introduce a sixth category for the application of NLP in radiation oncology–namely, to assist in the analysis of radiotherapy incident reports. Specifically,, we present an attempt to use NLP to automate the classification of incident reports by the process step, problem type, and contributing factors.

Objective

The objective of this research project was to demonstrate the usefulness of NLP as a tool for incident learning in radiation therapy. Specifically, we aim to use supervised learning to rank the labels for the incidents by likelihood, and to use unsupervised learning to group similar incidents together, and visualize the incidents and understand which kinds of incidents recur the most.

Methods

Data Preprocessing

We obtained the narratives and expert-generated labels of 734 and 2,298 radiation therapy incidents from the McGill University Health Centre (MUHC) and the Canadian Institute for Health Centre (CIHI), respectively. The MUHC had labelled their incidents for 38 fields of the NSIR-RT, while the CIHI for only 4: process step, problem type, contributing factors, and overall severity. Thus, we helped label incidents by only these 4 fields. In addition, the MUHC labelled their incidents with the 2015 version of the NSIR-RT, while the CIHI with the 2017 version, which combined some old labels into one and added some new labels. Thus, we mapped the old values to the new to combine the data. If an old label maps to several new labels, we preserved the old label.

We cleaned the incident narratives by first removing line breaks and redundant spaces, isolating numerals and punctuations, and lowercasing the reports. We detected and attempted to translate 20 French reports to English using the *Google Translate API* (Patel 2014), which preserves abbreviations and misspellings. However, we were unhappy with the translated texts and ultimately translated them ourselves. Next, we resolved abbreviations among words shorter than five characters with the help of a content expert [JK], and autocorrected misspellings using the *Enchant* spellchecker (Enchant 2010). Abbreviations were identified by selecting out all words shorter than five characters that were not stop words, which are commonly used words such as “an” and “the.”. Afterwards, we used *Metamap* (Aronson 2010), to normalize biomedical entities with their canonical forms in the *Unified Medical Language System* (UMLS) (Bodenreider 2004) and the *Radiation Oncology Ontology* (ROO). Furthermore, we normalized date entities recognized by the *spaCy* NLP library (Honnibal 2017) with the word "date," and similarly for percent, cardinal, quantity, and ordinal entities. Finally, we used spaCy again to lemmatize, which is to change a word to its lemma, the canonical form in which the word appears in a dictionary, and remove all stop words. An example of an MUHC-sourced incident narrative and its cleaned form are provided in Table 1:

Original Narrative Text	Cleaned Narrative Text
INCCURATE TARGET. CTV WAS BIGGER THAN PTV, WAS NOTICED ONLY AT THE END OF PLANNING PROCESS, TARGET HAD TO BE CORRECTED AND PLAN REDONE.	inaccurate target ctv be big than ptv be noticed only at the end of planning process target had to be correct and plan redone

Table 1. Example of an MUHC-source incident narrative and its cleaned form. The spellchecker recognized and corrected the underlined word.

Data Vectorization

Data vectorization is the process of transforming data into vectors or matrices that may be input to machine learning algorithms. We vectorized the expert-generated labels with one-hot encoding, and the preprocessed text with count vectorization, term frequency-inverse document frequency (TF-IDF) vectorization, and *fastText* word embedding (Wu 1992). Each of these is described below.

One-hot Encoding

We vectorized the expert-generated labels into a one-hot encoding matrix Y by assigning 1 to $Y_{i,j}$, the cell at row i and column j, if the incident report i was labelled with label j, and 0 otherwise. Scikit-learn (Pedregosa 2011)'s *OneHotEncoder* function was used to automate the process.

Count Vectorization

We vectorized the preprocessed incident narratives into a document term matrix X by assigning n to $X_{i,j}$ if word j occurs n times in incident report i. The *Tm* package in *R*, specifically the *DocumentTermMatrix* function, was used to automate the vectorization.

TF-IDF Vectorization

We vectorized the preprocessed incident reports into a TF-IDF matrix X by assigning $X_{i,j}$ a TF-IDF weight $tf_{i,j} \cdot ln\,(\frac{N}{df_i})$, where $tf_{i,j}$ is the number of times word i occurred in incident report j, N the total number of incident reports, and df_i the number of incident reports containing word i. *Scikit-learn*'s *TfidfVectorizer* automated the process.

Word Embedding

We vectorized each preprocessed incident report into an embedding matrix by using *fastText's skip-gram* model to map words to vectors, then assigning the vector of the first word to the first column of the embedding matrix, and so on. Keras's *Tokenizer* and *Embedding* was used for automation.

Supervised Learning

In order map an incident report to a label, we assembled and compared over 500 multi-output ensemble regressors, the components of which were available in scikit-learn. As illustrated in Figure XX, to narrow down the selection of regressors we used a three-iteration cross validation process.

Figure XX

We first partitioned the incidents into three subsets for cross-validation. In each of the three iterations of the cross validation process, one subset was used as the testing set and the remainder as the training set. In each iteration, we used the rows of the TF-IDF and one-hot encoding matrices, representingthe incidents in the training set as the training predictors and training targets, respectively. We then obtained the testing predictors and testing targets similarly. During cross-validation, the multi-output regressors learned to map the training predictors to the training targets, and we compared the mapping of the testing predictors with the testing targets.

We then extended scikit-learn's 52 regressors to 104 multi-output regressors with scikit-learn's MultiOutputRegressor and RegressorChain. MultiOutputRegressor fitted a regressor per label in parallel, whereas RegressorChain did so in series such that the predictions from previous regressors are used as predictors. We then extended the 104 multi-output regressors into 520 multi-output ensemble regressors with scikit-learn's BaggingRegressor, ExtraTreesRegressor, RandomForestRegressor, AdaBoostRegressor, GradientBoostingRegressor. The first three are variations of bagging, which is averaging the output of the multi-output regressors fitted on the disjoint subsets of the training set. The last two are variations of boosting, which is fitting regressors in series such that each regressor corrects the mistakes of the preceding one. We thus assembled 104 multi-output regressors and 520 multi-output ensemble regressors.

We next cross-validated the multi-output regressors with scikit-learn's `cross_validate` and `make_scorer` which used in cross-validation our custom function that scored the multi-output regressors based on how close to the top of the dropdown menu they placed the expert-generated labels on average. In addition, we limited the cross-validation of each multi-output regressor to 3-5 minutes based on the number of labels in the field, as we sought to tune only the multi-output regressors that were reasonably fast. We afterwards tuned the 10 best performing multi-output regressors with grid-search, the scoring and comparing of an exhaustive set of parameters in search of the best parameter set. This is automated

by scikit-learn's GridSearchCV, and we repeated this process for each process step, problem type, contributing factors, and overall severity.

Take process step for example. Let X be the TF-IDF matrix, in which row i, denoted $X_{i,*}$ is the vector that represents incident report i; and Y be the one-hot encoding matrix for process step, in which each column is a label for process step, and $X_{i,j}$ is 1 if incident i belongs to label j, and 0 otherwise.

Before training and testing a regressor, we first split the indices of the incident reports into a training set A and a testing set B. We then denote the rows of the TF-IDF and one-hot encoding matrices in the training set as X_A and Y_A, respectively. We similarly denote the rows of the TF-IDf and one-hot encoding matrices in the testing set as X_B and Y_B, respectively. Finally, since process step has 10 labels, we denote the 10 columns of Y_A as $y_1, y_2, ..., y_{10}$.

Scikit-learn offers over 50 single-output regression algorithms. Say we are training algorithm f on X_A and y_j, then we denote the regressor as f_{X_A,y_j}. This regressor would then take any row of a TF-IDF matrix, for example $X_{i,*}$, and output a real number. Since X_A are spanned by 0s and 1s, the output is likely between 0 and 1. We can then interpret the output as the probability that incident report i belongs to label j.

Since we have 10 labels for process step, we would have trained $f_{X_A,y_1}, f_{X_A,y_2}, ..., f_{X_A,y_{10}}$. We can then combine them into a multi-output regressor F_{f,X_A,Y_A}, defined by $F_{f,X_A,Y_A}(X_{i,*}) = (f_{X_A,y_1}(X_{i,*}), f_{X_A,y_2}(X_{i,*}), ..., f_{X_A,y_{10}}(X_{i,*}))$ for any row $X_{i,*}$ in X. We can interpret this vector of 10 real numbers as the probabilities that incident i belongs to each of the 10 labels, and then order the labels by decreasing likelihood to facilitate labelling.

We next devise a custom scoring function g to evaluate F_{f,X_A,Y_A} by comparing the output of F_{f,X_A,Y_A} given X_B as input to Y_B. For any incident report $X_{B_{i,*}}$ in the X_B, let $Y_{B_{i,*}} = (a_1, a_2, ..., a_{10})$ be the expert-generated labels, and $F_{f,X_A,Y_A}(X_{B_{i,*}}) = (b_1, b_2, ..., b_{10})$ be the predicted labels. We let g output the lowest placement of any expert-generated label in the reordering. In other words, let σ be the permutation defined by $\sigma(j) = k$ if b_j is the kth greatest value in $(b_1, b_2, ..., b_{10})$, and

$I = \{j \in \{1, 2, ..., 10\} : a_j = 1\}$ the index set of labels marked true by the experts. Then, $g(F_{f,X_A,Y_A}(X_{B_{i,*}}), Y_{B_{i,*}}) = argmax_{j \in I} \sigma(j)$.

Finally, to evaluate the overall performance of the algorithm, we let G output the average of g on each incident report in B. In other words, $G(f, X, Y, A, B) = \frac{1}{|B|} \sum_{i \in B} g(F_{f,X_A,Y_A}(X_{B_{i,*}}), Y_{B_{i,*}})$. Note that the less this value, the better the performance of the algorithm.

In implementing the above architecture, we reduce the bias of the splitting of the data into A and B with a 3-fold cross-validation. In this process, we split the indices of the incident reports into three equal index sets or folds, say K_1, K_2, K_3. For three iterations, we take one fold as B, and the rest as A. Finally, we average the output of $G(f, X, Y, A, B)$ in the three iterations, and use this average as a better evaluation of f. We also cap the time limit for cross-validation to a few minutes to ensure that the algorithms can be later tuned in a reasonable time frame.

Scikit-learn offers 52 single-output regressors, which we denote as S_f; 2 multi-output regressors or ways to assemble F_{f,X_A,Y_A} from $f_{X_A,y_1}(X_{i,*}), f_{X_A,y_2}(X_{i,*}), ..., f_{X_A,y_{10}}(X_{i,*})$, which we denote as S_F, and 5 ensemble methods, which are bagging and boosting techniques that reduce variance and bias, respectively, to use with any f. We denote using ensemble method b with f as $b \circ f$, and the set as S_b. We redefine G to include the variability of f, F, and b.

$G(F, b, f, X, Y, A, B) = \frac{1}{|B|} \sum_{i \in B} g(F_{b \circ f,X_A,Y_A}(X_{B_{i,*}}), Y_{B_{i,*}})$. Then, the below pseudocode and legend sum up this section.

Cross-Validation (F, b, f, X, Y)

Partition indices into K_1, K_2, K_3 of equal cardinality

While within time limit **do**

Try

Return $\frac{1}{3}\sum_{k=1}^{3} G(F, b, f, X, Y, \cup_{j\neq k} K_j, K_k)$

Catch

Return ∞

Return ∞

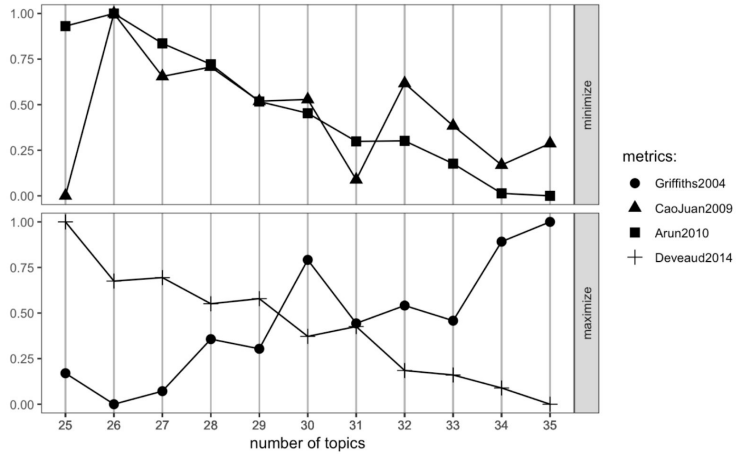
Best-Multi-Output-Ensemble-Regressor $(S_F,\ S_b,\ S_f, X, Y)$

Return $argmin_{F,b,f}$ **Cross-Validation** (F, b, f, X, Y)

Unsupervised Learning

Unsupervised learning is modelling the structure of unsorted data, such as clustering unlabelled incident reports into topics, which can then be interpreted as possible problem types or contributing factors.

We first used the document term matrix to build latent Dirichlet allocation models of different number of topics, and selected the model that minimized the metrics by Arun et al. and Juan et al, and maximized those by Deveaud et al. and Griffiths et al, as shown in the following figure. Ldatuning's FindTopicsNumber and FindTopicsNumber_plot automated this, and we captured the model in a JSON file with LDAvis's createJSON.



Multi-metric grid-search reveals 30 possibly new problem types or contributing factors in the incident descriptions labelled as "other."

We then explored the model in a shareable interactive visualization created from the JSON file by LDAvis's serVis. The visualization revealed which topics, or types of incidents recurred the most in the left panel, a two-dimensional projection of the topics where proximity indicated similarity, and area indicated frequency. The visualization also revealed how the topics, or types of incidents, related to each other in the right panel, which comprised a ranking of the keywords by their relevance to a selected topic, such that selecting a keyword would highlight the topics linked together by the keyword. The visualization can be exported and shared as a web application with SimpleHTTPServer.

Finally, we repeated this process for incident reports labelled as "other" for each of process step, problem type, and contributing factors. We examined the topics, each of which may represent a recurring incident to which no label applies, as possible candidates for new labels.

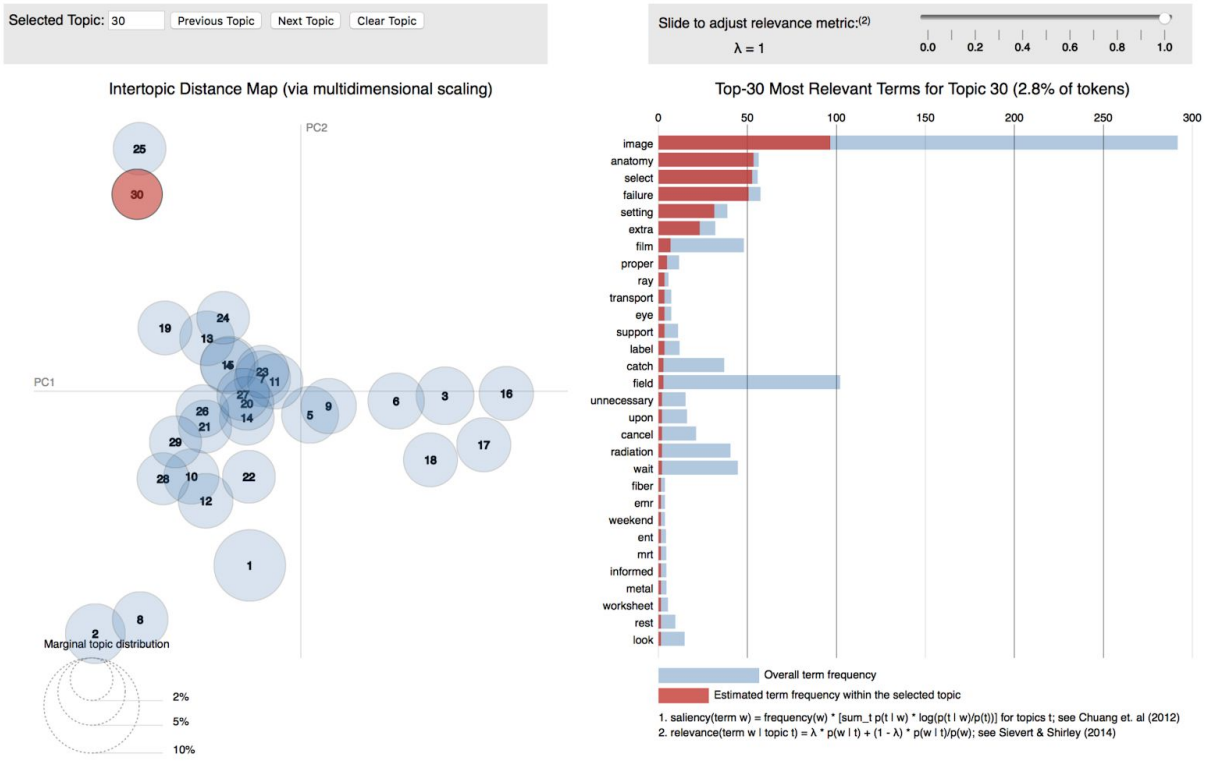


Figure 3: Topic 30 reveals that extra images were taken due to failure to select anatomy settings.

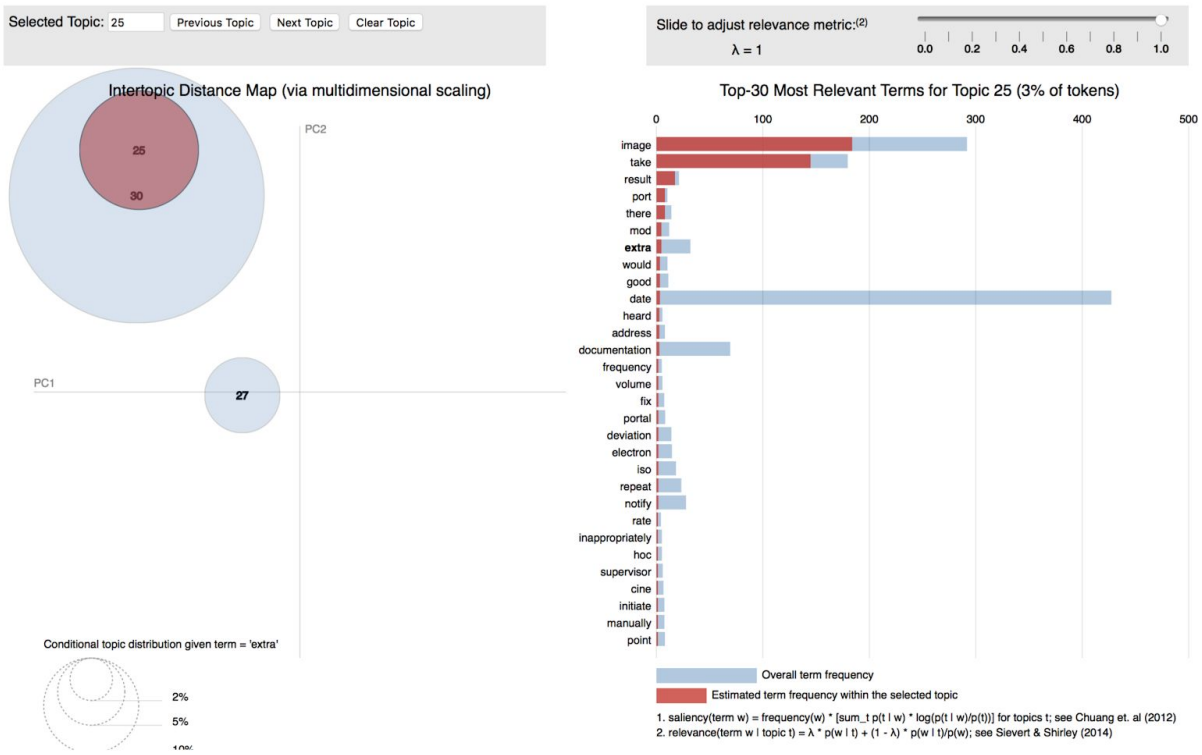


Figure 4: Topic 25, which shares the word "extra" with topic 30, reveals that extra port images were also taken.

Results

Data Preprocessing

In the tables below, the labels for each of process step, problem type, and contributing factors are displayed beside the percentage of incident reports tagged with the label.

Process Step	
Treatment delivery	34.10
	20.28
Contouring and planning	18.54
Imaging for treatment planning	13.39
Patient medical consultation and physician assessment	6.33
Pre-treatment quality assurance	4.22
Radiation treatment prescription scheduling	1.48
On-treatment quality assurance	1.06
Interventional procedure for planning and/or delivery	0.33
Post-treatment completion	0.26

Problem Type	
	39.05
Wrong patient position, setup point, or shift	15.77
Radiation therapy scheduling error	10.95
Wrong, missing, mislabeled, or damaged treatment accessories	9.76
Excess imaging dose	3.36
Systematic hardware/software (including dose-volume) error	3.10
Wrong target or OAR contours or wrong planning (Retired Value)	2.51
Wrong prescription dose-fractionation or calculation error	1.91
Failure to perform on-treatment imaging as per instructions	1.65
Inadequate coordination of combined modality care	1.39
Wrong anatomical site (excluding laterality)	1.39
Wrong patient	1.32
Wrong side (laterality)	1.29
Fall or other patient injury or medical condition	1.25
Wrong target or OAR contours	0.99
Untimely access to medical care or radiotherapy	0.79
Treatment plan acceptable but not physically deliverable	0.79
Wrong plan dose (Retired value)	0.69
Interventional procedure error (Retired value)	0.59
Inappropriate or poorly informed decision to treat or plan	0.46
Treatment not delivered - personnel/hardware/software failure	0.33
Treatment plan (isodose distribution) unacceptable	0.33
Wrong planning margins	0.16
Infection	0.13
Allergic reaction	0.03

Contributing Factors	
Policies and/or procedures not followed	19.69
Distraction or diversions involving staff	14.47
Expectation bias involving staff	13.17
Communication or documentation inadequate (patient specific)	12.42
Staff behaviour	9.63
Policies and/or procedures non-existent or inadequate	8.09
	7.17
Equipment software or hardware design, including 'human factors' design, inadequate	3.89
Failure to identify potential risks	3.83
Patient or family member medical condition, preference or behaviour	1.43
Staff education or training inadequate	1.26
Change management	1.10
Handoffs inadequate	0.70
Human resources inadequate	0.64
Equipment quality assurance and/or maintenance inadequate	0.62
External factors beyond programmatic control	0.52
Organizational and/or workspace resources inadequate (excluding human resources)	0.38
Equipment software or hardware commissioning, calibration or acceptance testing inadequate	0.36
Unfamiliar treatment approach or radiation treatment technique	0.34
Patient or family member medical condition preference or behaviour	0.14
Patient education inadequate	0.12

Bagging is the best ensemble method across the board, and multi-output regression consistently perform better than regressor chain. The best multi-output bagged regressors after tuning for sorting process step, problem type, and contributing factors, were support vector regressor $(G = 1.57)$, passive aggressive regressor $(G = 2.36)$, and random forest regressor $(G = 5.25)$, respectively. In other words, they place the expert-generated labels on average 1.57th, 2.36th, and 5.25th among the 9 process step, 24 problem type, 21 contributing factors labels, respectively.

Below are the performance of the top regression algorithms on sorting the labels of each of process step, problem type, and contributing factors.

	Ridge	Linear SVR	Passive Aggresive Regressor	Random Forest Regressor
Process Step	3.53	1.57	1.58	1.70
Problem Type	6.84	2.41	2.36	2.85
Contributing Factors	N/A	5.42	5.39	5.25

Unsupervised Learning

Unsupervised learning algorithm clustered the 3162 incidents into 44 topics, and the 634, 1275, and 333 incidents labelled as "other" for process step, problem type, and contributing factors into 24, 30, and 27 topics, respectively.

Below are the top 5 topics for each of process step, problem type, and contributing factors, along with their top 5 keywords of each topic, and the beta value which indicates the importance of the keyword to the topic.

Process Step			Problem Type			Contributing Factors		
topic	term	beta	topic	term	beta	topic	term	beta
1	ã¢	0.07	1	ordinal	0.52	1	plan	0.30
1	ã£	0.06	1	treatment	0.01	1	time	0.11
1	the	0.04	1	couch	0.01	1	ready	0.11
1	issue	0.03	1	rotation	0.01	1	come	0.06
1	correct	0.03	1	beam	0.01	1	task	0.01
2	cardinal	0.12	2	cardinal	0.44	2	req	0.10
2	instead	0.03	2	prescriptio	0.05	2	scan	0.06
2	use	0.01	2	field	0.04	2	site	0.03
2	find	0.01	2	course	0.03	2	approve	0.03
2	departmer	0.01	2	phase	0.00	2	change	0.03
3	the	0.08	3	patient	0.48	3	cardinal	0.46
3	new	0.05	3	treat	0.05	3	arc	0.04
3	deliver	0.02	3	start	0.02	3	apron	0.00
3	contour	0.02	3	need	0.00	3	dosimetry	0.00
3	error	0.01	3	open	0.00	3	work	0.00
4	patient	0.51	4	task	0.16	4	fall	0.04
4	have	0.00	4	send	0.11	4	need	0.04
4	physics	0.00	4	review	0.07	4	head	0.02
4	cancer	0.00	4	cosimo	0.03	4	right	0.02
4	report	0.00	4	contour	0.03	4	come	0.02
5	call	0.06	5	have	0.13	5	time	0.19
5	appt	0.06	5	new	0.04	5	see	0.10
5	floor	0.04	5	dosimetry	0.02	5	ptn	0.08
5	cosimo	0.04	5	machine	0.02	5	wait	0.07
5	porter	0.03	5	the	0.01	5	long	0.03

Discussion

Supervised Learning

We first summarize here the workflow of using supervised learning to facilitate incident learning. First, we require a dataset of incidents for algorithms to learn from, we must therefore report and label incoming incidents to accumulate such a dataset. Then, we implement supervised learning algorithms as detailed in the methods section. The algorithms that are both efficient and effective for natural language processing tasks such as this are support vector regression, passive aggressive regression, random forest regression, and long short-term memory recurrent neural network. We can implement any one of them without bagging or boosting as a starting point. Next, we feed new incident descriptions into the algorithm, which then sorts the labels for process step, problem type, and contributing factors. If the incident reports are too numerous to label, we can then trust the recommendations made by the algorithm. Otherwise, sorting the labels significantly decreases the time to label incoming incidents, as experts no longer need to choose from more than some 20 labels.

The above method of ranking labels for text by likelihood is generic, that is, we can apply it to any set of labels and any dataset of reports. For instance, if the reports are medical reports, and the labels are diagnoses, then the above process produces a program that suggests diagnoses for medical reports. If the reports are medical reports, and the labels are courses of treatment, then the above process produces a program that suggests courses of treatment for medical reports. Since the program merely ranks and suggests, and humans make the final decision, we speed up the process while not compromising quality. On a side note, since medical reports are filled with a relatively smaller set of jargons compared to the vocabulary of the incident reports which are miscellaneous, machine learning methods likely perform better on medical reports than incident reports. The main contribution of this method is injecting machine learning into the workflow such that humans remain in control.

Unsupervised Learning

We again first summarize the workflow for using unsupervised learning to facilitate incident learning. Suppose the incoming incident reports exceed the human resources to label them one by one, unsupervised learning can automatically group similar incident reports together without any human effort. We can then visualize the topics as shown in the methods section, determine which groups are the largest, and quickly infer from the keywords the kinds of incidents in these groups, and therefore the kinds of the incidents that recur the most, which we can understand further by reading the individual incident reports. We thus have a priority list of incidents to address based on their frequency of recurrence. Finally, we address these incidents by improving protocols, removing hazards, and so on.

There are several advantages of using unsupervised learning over the supervised learning to learn from incidents. First, unsupervised learning does not require a painstakingly labeled dataset, only a dataset of incident descriptions. Second, unsupervised learning is almost instant, requiring no human input in presenting the incidents in a format that makes clear the changes that we must make, and in what order. Third, supervised learning works with a set of existing labels, and cannot discover new kinds of incidents outside of the existing labels. In unsupervised learning, we can discover these kinds of incidents as we survey the groups, and possibly later add new labels to the existing set. Fourth, since unsupervised learning requires no human effort, except during the last stage during which we interpret the results, we can take into account much more than personnel-generated incident reports. Specifically, with the advent of institution-specific patient portal mobile applications, such as the Opal app of McGill University Health Centre, we can collect patient-generated incident reports, and examine how best to improve radiotherpay from the patients' perspective. It is likely that this patient-centric approach to incident learning can yield new insights that improve both patient safety and comfort. Finally, as the supervised learning approach above, this unsupervised approach is generic, applicable to incident learning in any medical department.

Conclusion

Supervised and unsupervised learning can significantly facilitate incident learning in radiotherapy and beyond.

Citations

Punit Kaur, Mark D. Hurwitz, Sunil Krishnan, Alexzander Asea. Combined Hyperthermia and Radiotherapy for the Treatment of Cancer. Cancers 3, 3799–3823 MDPI AG, 2011.

Logan Montgomery. An evaluation of incident learning using the taxonomy of the Canadian National System for Incident Reporting - Radiation Treatment. (2016).

E. C. Ford, L. Fong de Los Santos, T. Pawlicki, S. Sutlief, P. Dunscombe. Consensus recommendations for incident learning database structures in radiation oncology. Medical Physics 39, 7272–7290 Wiley-Blackwell, 2012.

A. Mans, M. Wendling, L. N. McDermott, J.-J. Sonke, R. Tielenburg, R. Vijlbrief, B. Mijnheer, M. van Herk, J. C. Stroom. Catching errors with in vivo EPID dosimetry. Medical Physics 37, 2638–2644 Wiley-Blackwell, 2010.

Cunningham, Joanne, et al. "Radiation Oncology Safety Information System (ROSIS)–profiles of participants and the first 1074 incident reports." Radiotherapy and Oncology 97.3 (2010): 601-607.

Huq, M. Saiful, et al. "The report of Task Group 100 of the AAPM: Application of risk analysis methods to radiation therapy quality management." Medical physics 43.7 (2016): 4209-4262.

Aronson, Alan R., and François-Michel Lang. "An overview of MetaMap: historical perspective and recent advances." Journal of the American Medical Informatics Association 17.3 (2010): 229-236.

Bodenreider, Olivier. "The unified medical language system (UMLS): integrating biomedical terminology." Nucleic acids research 32.suppl_1 (2004): D267-D270.

Honnibal, Matthew, and Ines Montani. "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing." To appear (2017).

Wu, Sun, and Udi Manber. "Fast text searching allowing errors." Communications of the ACM 35.10 (1992): 83-92.

Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

Meystre, Stéphane M., et al. "Extracting information from textual documents in the electronic health record: a review of recent research." Yearb Med Inform 35.8 (2008): 128-144.

Patil, Sumant, and Patrick Davies. "Use of Google Translate in medical communication: evaluation of accuracy." Bmj 349 (2014): g7392.

Pons, Ewoud, et al. "Natural language processing in radiology: a systematic review." Radiology 279.2 (2016): 329-343.

Michael Milosevic, Crystal Angers, Brian Liszewski, C. Suzanne Drodge, Eve-Lyne Marchand, Jean Pierre Bissonnette, Erika Brown, Peter Dunscombe, Jordan Hunt, Haiyan Jiang, Krista Louie, Gunita Mitera, Kathryn Moran, Tony Panzarella, Matthew Parliament,

Spencer Ross, Michael Brundage. The Canadian National System for Incident Reporting in Radiation Treatment (NSIR-RT) Taxonomy. Practical Radiation Oncology 6, 334–341 Elsevier BV, 2016.

"Enchant." AbiWord: Word Processing for Everyone, www.abisource.com/projects/enchant/.

J. Jehring, H. W. Heinrich. Industrial Accident Prevention: A Scientific Approach. Industrial and Labor Relations Review 4, 609 JSTOR, 1951.