Journal of the American Medical Informatics Association, 0(0), 2017, 1–10
doi: 10.1093/jamia/ocx131
Research and Applications

Research and Applications

Hierarchical attention networks for information extraction from cancer pathology reports

Shang Gao,[1] Michael T Young,[1] John X Qiu,[1] Hong-Jun Yoon,[1] James B Christian,[1] Paul A Fearn,[2] Georgia D Tourassi,[1,*] and Arvind Ramanthan[1,*]

[1]Computational Science and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA and [2]Surveillance Informatics Branch, Division of Cancer Control and Population Sciences, National Cancer Institute, Bethesda, MD, USA

[*]Corresponding Author: Arvind Ramanathan, Computational Sciences and Engineering Division, Health Data Sciences Institute, Oak Ridge National Laboratory, MS-6085, One Bethel Valley Road, Oak Ridge, TN 37831-6085, USA. Email: ramanathana@ornl.gov. Phone: 865-576-7266. Fax: 865-241-0337

Received 12 June 2017; Revised 10 October 2017; Editorial Decision 15 October 2017; Accepted 26 October 2017

ABSTRACT

Objective: We explored how a deep learning (DL) approach based on hierarchical attention networks (HANs) can improve model performance for multiple information extraction tasks from unstructured cancer pathology reports compared to conventional methods that do not sufficiently capture syntactic and semantic contexts from free-text documents.

Materials and Methods: Data for our analyses were obtained from 942 deidentified pathology reports collected by the National Cancer Institute Surveillance, Epidemiology, and End Results program. The HAN was implemented for 2 information extraction tasks: (1) primary site, matched to 12 International Classification of Diseases for Oncology topography codes (7 breast, 5 lung primary sites), and (2) histological grade classification, matched to G1–G4. Model performance metrics were compared to conventional machine learning (ML) approaches including naive Bayes, logistic regression, support vector machine, random forest, and extreme gradient boosting, and other DL models, including a recurrent neural network (RNN), a recurrent neural network with attention (RNN w/A), and a convolutional neural network.

Results: Our results demonstrate that for both information tasks, HAN performed significantly better compared to the conventional ML and DL techniques. In particular, across the 2 tasks, the mean micro and macro *F*-scores for the HAN with pretraining were (0.852,0.708), compared to naive Bayes (0.518, 0.213), logistic regression (0.682, 0.453), support vector machine (0.634, 0.434), random forest (0.698, 0.508), extreme gradient boosting (0.696, 0.522), RNN (0.505, 0.301), RNN w/A (0.637, 0.471), and convolutional neural network (0.714, 0.460).

Conclusions: HAN-based DL models show promise in information abstraction tasks within unstructured clinical pathology reports.

Key words: clinical pathology reports, information retrieval, recurrent neural nets, attention networks, classification

OBJECTIVE

Cancer is the second leading cause of death in the United States.[1] Given the public health burden that cancer imposes on society, developing effective clinical surveillance of cancer remains one of the top priorities for the National Cancer Institute (NCI). The NCI Surveillance, Epidemiology, and End Results (SEER) program maintains a comprehensive database of diagnostic, treatment, and outcomes information from vast amounts of unstructured data sources such as clinical visit notes, pathology reports, treatment summaries, and so on. However, a critical challenge in leveraging these documents in the context of cancer surveillance is abstracting essential

information regarding tumor type, location, and histological grade for a given patient at a given time.

The primary objective of our study is the design, implementation, and validation of a novel deep learning (DL) approach based on hierarchical attention networks (HANs) to automate information extraction from unstructured clinical pathology reports. We demonstrate that HANs can effectively capture the primary information regarding a patient's tumor type, grade, and location from unstructured text. Compared to traditional natural language processing (NLP) approaches, we show that HANs can enhance the efficiency and quality of data extraction across 2 distinct information extraction tasks: identifying the primary site and the histological grade of a tumor from pathology reports. Taken together, our approach opens new opportunities for scalable and automatic information extraction from unstructured clinical pathology reports.

BACKGROUND

A key goal of the NCI SEER program is to provide a comprehensive measurement of cancer incidence, morbidity, survival, and mortality for persons with cancer.[2] For this purpose, the SEER program relies on the acquisition of diagnostic, treatment, and outcomes information through manual information abstraction and information processing by expert staff from vast amounts of unstructured sources (eg, clinical visit notes, pathology reports, treatment summaries, etc.). This manual processing of information imposes inherent limitations on the volume and types of information that can be collected. Furthermore, the reports can be highly variable because they are sourced from hundreds of different health care providers and across multiple laboratories. Additional variability is caused by human factors such as different interpretations of coding rules and human error. Thus, manual classification of cancer pathology reports can be a challenging and time-consuming task that requires extensive training. With the continued growth in the number of cancer pathology reports, the rise in patient survival rates, and the increase in treatment complexity, cancer registries face a significant challenge in manually reviewing large volumes of reports.

One solution to this problem currently being explored is the application of artificial intelligence and machine learning (ML) to automatically read and extract information from cancer pathology reports. If effective, such a solution would significantly reduce delays in report processing and allow trained personnel to focus their time on analysis and research. However, the content in cancer pathology reports can be difficult for machines to process due to its high variability, including misspellings and missing punctuation, clinical diagnoses interspersed with complex explanations, different terminology to label the same cancer, and information about multiple cancers included in a single report. Developing an automated solution with high accuracy and consistency across a wide selection of reports is therefore challenging.

Whereas traditional ML models require human experts to encode domain knowledge through feature engineering, DL approaches are able to learn salient feature representations through back-propagation on the given dataset. This makes DL especially apt for NLP tasks where manual encoding of features is both inefficient and impractical. Recent experiments have shown that DL approaches that generate their own features can achieve state-of-the-art results on NLP tasks such as question answering, part-of-speech tagging, and sentiment analysis.[3] Convolutional neural networks (CNNs), traditionally used for computer vision, have also been adapted for NLP tasks.[4] In computer vision, CNNs use a sliding window of learned filters to identify the important features in an image. In NLP, the filters of a CNN are instead trained to process segments of words to identify the word combinations that are most pertinent to a particular task.

Recurrent neural networks (RNNs) have been shown to be especially effective in NLP tasks thanks to their specialized architecture for processing time-series data.[5] Unlike traditional feed-forward neural networks, which take in an input and produce an output, RNNs are designed to take in a series of inputs over time. The output of a recurrent neural network depends on not only the input at the current timestep, but also inputs in previous (and possibly future) timesteps. In NLP tasks, RNNs process one word at a time and then learn linguistic patterns based on different sequences of words.

In basic RNNs, input data are written into the RNN at every timestep, even if they are irrelevant to the task at hand. This results in a dilution effect over time, so basic RNNs are unable to retain information and find relationships over a large number of timesteps. In NLP, this is problematic when semantic meaning is distributed over a long sequence of words. Long short-term memory (LSTM) cells[6] and gated recurrent units (GRUs)[7] are RNN architectures that address this problem by using gating functions that control the flow of information into and out of the RNN. These architectures selectively process and output information based on its relevance – inputs that constitute noise can be ignored, and important inputs can be stored and saved within the RNN until they are required (for details, see Supplementary Information S2–3). In NLP, LSTM- and GRU-based architectures are often used in question-answering tasks in which semantic meaning is spread across multiple sentences.[3]

Attention mechanisms can significantly boost the performance of RNNs by allowing them to further focus on timesteps that are most critical to the task at hand.[8] In regular RNNs, decisions are made simply by using the RNN output at the final timestep; this means the RNN must capture the essence of the entire input sequence in a single output vector, which is not always effective. With attention, the RNN saves an output at every timestep, and the attention mechanism then selects and combines the most important outputs based on their relevance to the task. This gives the RNN far more expressive power by allowing it to save useful information at every timestep and then later choose which outputs to use for decision-making.

Yang and co-workers[9] developed a HAN for classification of Yelp and movie reviews that outperformed both RNNs and CNNs. HANs expand on RNNs by utilizing hierarchies of RNNs that process long texts by breaking them down into smaller, more manageable segments. In this paper, we show that a similar architecture can be applied to effectively classify cancer pathology reports in different classification tasks.

Approaches for automated document classification in biomedical informatics range from carefully crafted rule-based systems[10] to traditional machine learning classifiers based on hand-engineered features[11] to deep learning approaches that require no manual feature engineering.[12] Jouhet and colleagues[13] tested the effectiveness of naïve Bayes and support vector machine (SVM) using term frequency–inverse document frequency (TF-IDF) features on classifying the International Classification of Diseases for Oncology (ICD-O-3) topographical codes and the morphological axes of the ICD-O-3 codes for cancer pathology reports. Jagannatha and Yu[14] showed that RNNs using LSTM or GRU units can effectively classify medical events from unstructured text in electronic health record notes. Qiu and co-authors[15] demonstrated that CNNs can classify the ICD-O3 topographical codes with higher accuracy than naive Bayes, SVM, and other traditional ML classifiers.

In this paper, we use 2 cancer pathology report classification tasks to demonstrate that HANs outperform not only traditional ML approaches, but also other DL architectures in the 2 classification tasks.

MATERIALS AND METHODS

Dataset description and preprocessing

The SEER pathology reports are obtained from 5 cancer registries (Connecticut, Hawaii, Kentucky, New Mexico, and Seattle, Washington) with the proper institutional review board–approved protocol. The full corpus consists of 2505 deidentified reports. Some, but not all, of these pathology reports were manually annotated by cancer registry experts based on standard guidelines and coding instructions used in cancer surveillance, depending on the classification task of interest (see below). These annotations served as the labels for our information extraction tasks.

We focused on 2 information extraction tasks: (1) primary site and (2) histological grade classification. For the primary site classification task, we used a corpus of 942 deidentified pathology reports matched to 12 ICD-O-3 topography codes corresponding to 7 breast and 5 lung primary sites. For the histological grade classification task, we used a total of 645 labeled reports matched to 4 histological grades, G1–G4. For both tasks, we limited our training corpus to pathology reports with a single topography code and histological grade sourced only from the "Final Diagnosis" section of the report. In Table 1, we describe the corpus for the individual tasks.

We followed a standard preprocessing pipeline for each document of our corpus, illustrated in Supplementary Figure S1. The cancer pathology reports in our dataset are in extended markup language (XML) format. For each report, we removed all XML tags and used all text besides the identifier tags (registry ID, patient ID, tumor number, and document ID). Depending on the registry and the report, line breaks may or may not have been added to the body text to improve readability. In reports with line breaks, the line breaks are used to both demarcate segments of information (eg, line break after each tumor property) and break apart paragraphs of information after a set line length.

In cancer pathology reports for breast cancer, an important indicator for primary site is clock references (eg, 12 o'clock or 03:00) that indicate the location of the tumor on the breast.[16] However, there is very little consistency in the clock format across the corpus. To improve consistency across reports, we standardized all clock references to a single format: a number, 1–12, followed by the string "o' clock."

In addition, we set all alphabetical characters to lowercase and removed all non-alphanumeric symbols except for periods, colons, and semicolons, as these symbols are used to split long lines. To ensure that periods are only used to mark the ends of sentences, several additional preprocessing steps were taken to remove periods from abbreviations and floats; these steps are described in detail in our supporting information page, Supplementary 1–2. Lastly, we tokenized the final text.

Word embeddings

We used word embeddings to convert word tokens in each report into numerical vectors. Unlike simpler methods for representing words in vector form, such as bag-of-words and one-hot encoding, word embeddings represent the semantic meanings of words within the numerical vectors.[17] Words that are semantically similar are closer to each other in distance, while words that are semantically different are farther apart in distance. We evaluated 2 popular word embeddings, Word2Vec[18] and GloVe.[19]

Word2Vec is based on the premise that words that appear in the same context tend to be more semantically similar. Word2Vec creates word embeddings by using a feed-forward neural network to predict the neighboring words for a given word. Like Word2Vec, GloVe also uses word co-occurrence, but rather than predicting word context, GloVe is based on the idea that the ratio of the co-occurrence probabilities of 2 words can be used to embed semantic meaning for those words. We tested both Word2Vec and GloVe embeddings during hyperparameter optimization as well as different sizes for the word embedding vector. We trained word embeddings on our entire corpus of 2505 reports and converted each token to its corresponding embedding. This converted each report into a tensor of size $\boldsymbol{X}_{L\times w\times e}$, where $\boldsymbol{L}$ is the number of lines per report, $\boldsymbol{w}$ is the number of words per line, and $\boldsymbol{e}$ is the word embedding size. These tensors became the input into the HAN (Supplementary Figure S1).

Table 1. Distribution of labeled reports for the primary site classification task and the histological grade classification task

Primary site ICD-O-3 Topographical codes		
Code	Count	Description
C34.0	26	Main bronchus
C34.1	139	Upper lobe, lung
C34.2	11	Middle lobe, lung
C34.3	78	Lower lobe, lung
C34.9	191	Lung, NOS
C50.1	13	Central portion of breast
C50.2	36	Upper-inner quadrant of breast
C50.3	10	Lower-inner quadrant of breast
C50.4	63	Upper-outer quadrant of breast
C50.5	21	Lower-outer quadrant of breast
C50.8	62	Overlapping lesion of breast
C50.9	292	Breast NOS
Histological grades		
G1	124	Well differentiated (low grade)
G2	233	Moderately differentiated (intermediate grade)
G3	271	Poorly differentiated (high grade)
G4	17	Undifferentiated (high grade)

NOS, not otherwise specified.

Hierarchical attention network

HANs are a DL document classification model developed by Yang et al.[9] based on RNNs. They are composed of hierarchies in which the outputs of the lower hierarchies become the inputs to the upper hierarchies. This is based on the intuition that documents are composed of meaningful sequences of sentences, which are composed of meaningful sequences of words. Originally designed to classify reviews, HANs process documents one sentence at a time.[9] However, due to the structure and content of our pathology reports, the reports did not split naturally into sentences. We instead split each pathology report in our dataset with line breaks. To prevent long lines from reports without line breaks, we further split any line longer than 50 words with punctuation (periods, colons, and semicolons). This resulted in an average of 57 lines per document and 8 words per line for the entire corpus (Supplementary Figure S2).

Each hierarchy in the HAN is composed of a dynamic bidirectional LSTM or GRU with attention mechanisms (Figure 1). Bidirectionality is imposed so that the network can account for the preceding and following context when processing words/sentences.[5] LSTMs/GRUs are used because they allow the network to selectively process input information based on how relevant it is to the classification tasks; likewise, the attention mechanism is added to enable the network to put extra focus on the LSTM/GRU outputs associated with the words and lines that are most indicative of a particular

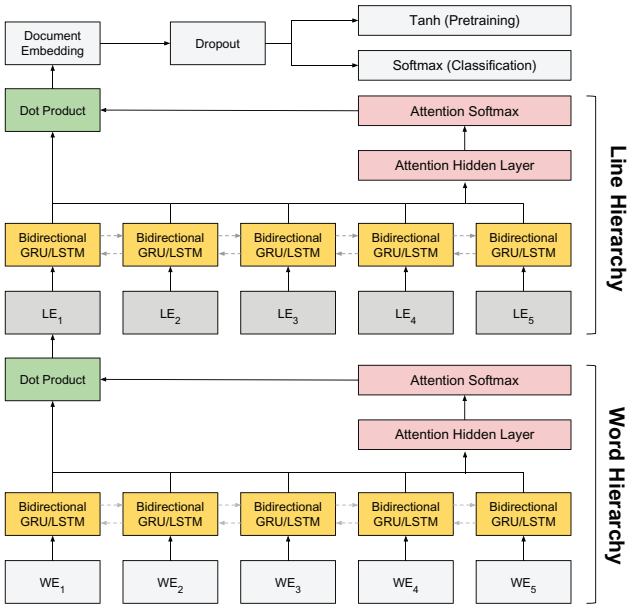

Figure 1. Architecture for our hierarchical attention network (HAN). produces line embeddings by processing the word embeddings in each line. The HAN then produces a document embedding by processing the line embeddings in the document. The final document embedding can then be used for classification or pretraining purposes.

class. The implementation details of our LSTMs, GRUs, and attention mechanism are available in our supplementary information, pages 2–3. LSTMs and GRUs generally have similar performance on most tasks, but in specialized tasks one may outperform the other.[20] We therefore tested both LSTMs and GRUs during hyperparameter optimization.

We modeled our HAN as 2 hierarchies. The lower hierarchy processes one line at a time, fed in as word embeddings. These are processed by a bidirectional LSTM/GRU with an attention mechanism that determines which words are most important. The output is a line embedding that captures the semantic content of the line. This is repeated for every line in a document. The upper hierarchy processes an entire document at a time by taking in the line embeddings generated from the lower hierarchy. These are fed into another bidirectional LSTM/GRU with an attention mechanism, all of which have the same architecture as those in the lower hierarchy. The final output is a weighted document embedding that represents the meaning of the entire document. We applied dropout to this final document embedding and then fed it into a softmax classifier to predict a class label.

Hyperparameter optimization

We used sequential optimization using gradient boosted trees to find the best hyperparameters for our HAN. This optimization method uses a gradient boosted tree-based regression model to predict the model performance at unexplored hyperparameter settings. We used this optimization method because tree-based optimization has been shown to converge faster than traditional Bayesian optimization.[21] The following hyperparameters were tuned: (1) type of word embeddings (Word2Vec or Glove); (2) size of word embeddings (100–500); (3) type of RNN unit used (GRU or LSTM); (4) number of hidden GRU or LSTM cells used in each bidirectional RNN layer (50–200); (5) size of hidden layer in attention mechanism (50–300); and (6) dropout on final document embedding (0.5 or none). We ran hyperparameter optimization separately for the primary site classification task and the histological grade classification task to find the best architecture for each task. We optimized our hyperparameters on a validation set; our procedure is described in detail in our supplementary information, page 3. The best hyperparameter setup for each classification task is listed in Supplementary Table S1.

Unsupervised pretraining step

Pretraining neural networks on unlabeled data using a modified autoencoder structure can improve the final performance of the network.[22] In these methods, the classification functions in the network are replaced with a decoder network that reconstructs the input data, allowing the network to learn inherent patterns in the data. While such autoencoder structures may be effective for short text segments, it is far more difficult for an autoencoder to accurately regenerate long documents such as pathology reports.[23]

We implemented a novel method of pretraining our HAN using unlabeled reports: we modeled the final layer before the softmax classification as a document embedding to succinctly represent the content within a document. We can pretrain the network to learn inherent patterns in documents by having it attempt to learn document embeddings that match some other unsupervised representation of document content. In our case, we trained our model to generate document embeddings to match the TF-IDF–weighted word embeddings of the corresponding document, as we found that these embeddings managed to capture some of the natural variation between documents of different classes (Supplementary Figure S3).

To pretrain our model, we first generated TF-IDF vectors on unigrams only for all nonempty pathology reports from the full corpus, a total of 2495 reports. We then normalized each TF-IDF vector to have a total sum of 1. We created TF-IDF–weighted word embeddings for a document by multiplying each index in that document's normalized TF-IDF vector with its corresponding word embedding and then summing the total. We normalized the resulting TF-IDF–weighted word embedding to have mean 0 and standard deviation 0.4, then clipped any values above 1 or below −1.

In our HAN, we replaced the softmax classification layer with a tanh layer that attempted to output the corresponding TF-IDF weighted word embedding for the document (Eqn. 1; Figure 1).

$$\text{pred}_i = \tanh(W_{\text{pred}}\text{emb}_i + b_{\text{pred}}) \quad (1)$$

The model was trained for 5 epochs using mean-square-error loss on all nonempty pathology reports from the full corpus (Eqn. 2), and the trained weights were saved and used to initialize all model weights during supervised classification.

$$\text{MSE} = \frac{1}{n}\sum_{i}^{n}(\text{pred}_i - \text{TFIDF}_i)^2 \quad (2)$$

Experimental setup and evaluation metrics

Our 2 classification tasks were primary site classification and histological grade classification. For both tasks, we used 10-fold cross-validation to evaluate model performance. For each fold, we split the dataset into 0.8/0.1/0.1 train/validation/test sets. We trained the HAN on the train set for 30 epochs and measured the performance on the validation set at each epoch. We saved the network architecture at the epoch with the highest validation performance, and then evaluated final model performance on the test set. Because of class imbalance, we used *F*-score as our evaluation metric for model performance. On the validation set, we chose the epoch with the highest

micro *F*-score, which is a weighted average of the *F*-score for each class label weighted by class size (Eqn. 3–5).

$$P^{\text{micro}} = \frac{\sum_{C_j}^{C} \text{TP}_j}{\sum_{C_j}^{C} (\text{TP}_j + \text{FP}_j)} \tag{3}$$

$$R^{\text{micro}} = \frac{\sum_{C_j}^{C} \text{TP}_j}{\sum_{C_j}^{C} (\text{TP}_j + \text{FN}_j)} \tag{4}$$

$$F^{\text{micro}} = \frac{2P^{\text{micro}}R^{\text{micro}}}{P^{\text{micro}} + R^{\text{micro}}} \tag{5}$$

For final model performance, in addition to measuring micro *F*-score, we also measured macro *F*-score, which simply averages the *F*-score of each class label regardless of class size (Eqn. 6).

$$F^{\text{macro}} = \frac{1}{|C|} \sum_{C_j}^{C} F(C_j) \tag{6}$$

Model baseline comparisons

We compared the performance of the HAN against the performance of several other predictive models that can be used for text classification. These include traditional ML classifiers as well as DL-based classifiers. For traditional ML classifiers, we tested naive Bayes, logistic regression, support vector machine, random forest, and XGBoost. We created input features for these traditional classifiers by removing punctuation and stop words from the pathology reports and then generating TF-IDF vectors on the resulting unigrams and bigrams with a minimum document frequency of 3. To tune each classifier, we applied the same hyperparameter tuning with gradient boosted trees that was used on the HAN on each of the traditional ML classifiers. The final hyperparameters used for each classifier are listed beneath the classifier names in Table 2. All classifiers were tested in Python using the *scikit-learn* package.

We also compared the performance of our HAN against the performance of recurrent RNNs and CNNs. For our comparison, we used the same word embedding vectors as we used for the HAN. Unlike the HAN, which takes in a document one sentence at a time, both the RNN and the CNN take in all word embeddings in the entire document at once.

We tested the performance of 2 RNN architectures. The first architecture was an RNN without an attention mechanism. This is equivalent to a single hierarchy of the HAN without the attention mechanism, a bidirectional RNN followed by dropout and softmax. The second architecture was an RNN with attention. This is equivalent to a single hierarchy of the HAN with the attention mechanism, a bidirectional RNN with attention followed by dropout and softmax. For both RNNs, we used the same optimized hyperparameters from our HAN hyperparameter optimization for each classification task. These baselines were designed to demonstrate how the attention mechanism and hierarchical structure of the HAN improve classification performance over basic RNNs. Furthermore, to confirm the necessity of the attention mechanism within both hierarchies of the HAN, we tested the performance of the HAN without the attention mechanism in either the word hierarchy or the line hierarchy.

For our CNN architecture, we used 3 parallel convolutional layers of 100 channels each with window sizes of 3, 4, and 5 words and stride of one word. For a given document, this results in an output of 300 channels × number of words. We then applied a maxpool to each of the 300 channels across all words in the document, resulting in a vector of length 300 for each document. A dropout of 50% was applied to this vector, which was finally fed into a softmax classifier. This particular architecture has been shown to perform well on cancer pathology report classification.[15]

To maintain consistency, all models were trained using the same 0.8/0.1/0.1 train/validation/test splitting as the HAN. All DL models were trained for 30 epochs, and the model weights during the epoch with the highest validation performance were used for test set evaluation.

EXPERIMENTAL RESULTS

Primary site classification

In the primary site classification task, a predictive model must predict the primary site of a cancer given the content of the cancer pathology report. The performance results on this task are displayed in Table 2, which includes 95% confidence intervals for each performance metric derived using bootstrapping.[24] The bootstrapping method is described in greater detail in our supplementary information, page 5.

We see that the HAN outperformed all other models in both the micro *F*-score metric, with a score of 0.800, and the macro *F*-score metric, with a score of 0.594. This was an improvement of 13% for micro *F*-score over the next best model, CNNs, and an improvement of 27% for macro *F*-score over the next best model, RNNs. Pretraining the network using TF-IDF–weighted word embeddings yielded increases in both the speed of convergence of the model and the final performance accuracy (Figure 2).

Histological grade classification

In the histological grade classification task, our model must predict the histological grade of a cancer given the cancer pathology report. The performance results on this task are displayed in Table 2, which includes 95% confidence intervals for each performance metric derived using bootstrapping.[24]

As in the primary site classification task, the HAN once again outperformed all other models on both the micro *F*-score metric, with a score of 0.916, and the macro *F*-score metric, with a score of 0.841. This was an improvement of 28% for micro *F*-score over the next best model, CNNs, and an improvement of 37% for macro *F*-score over the next best model, XGBoost. While pretraining the network using TF-IDF–weighted word embeddings yielded an initial increase in both the speed of convergence and the validation accuracy (Figure 2), pretraining did not boost the final test accuracy in the histological grade classification task. We believe this is because TF-IDF–weighted word embeddings do not capture the natural separation of classes for histological grades as well as for primary sites (Supplementary Figure S3).

Model visualizations

The HAN can provide additional insight about the corpus of pathology reports by utilizing the structure of its architecture. Given a document, we can use the attention mechanism weights in both the word and line hierarchies to visualize how much each word and each line contributes to the final classification of that document (Figure 3). We can also find the words in our vocabulary associated with the highest attention weights to identify the words that are the most important to each classification task (Supplementary Table 2).

In addition, because the final layer of the HAN creates a document embedding for each pathology report, we can use a

Table 2. Final test performance of classification models on each classification task

Traditional Machine Learning Classifiers				
Classifier	Primary site micro *F*-score	Primary site macro *F*-score	Histological grade micro *F*-score	Histological grade macro *F*-score
Naive Bayes	0.554 (0.521, 0.586)	0.161 (0.152, 0.170)	0.481 (0.442, 0.519)	0.264 (0.244, 0.283)
Logistic regression (penalty = l1, solver = liblinear, C = 3.3, multiclass = ovr)	0.708 (0.678, 0.737)	0.400 (0.361, 0.437)	0.657 (0.622, 0.693)	0.507 (0.433, 0.584)
Support vector machine (C = 25, kernel = sigmoid, gamma = 0.5, shrinking = true)	0.673 (0.643, 0.702)	0.396 (0.353, 0.435)	0.595 (0.558, 0.634)	0.472 (0.413, 0.540)
Random forest (num trees = 400, max features = 0.9)	0.701 (0.673, 0.730)	0.437 (0.406, 0.467)	0.694 (0.657, 0.727)	0.579 (0.503, 0.650)
XGBoost (max depth = 5, num trees = 300, learning rate = 0.3)	0.712 (0.683, 0.740)	0.431 (0.395, 0.466)	0.681 (0.643, 0.716)	0.612 (0.539, 0.673)
Deep Learning Classifiers				
Convolutional neural network	0.712 (0.680, 0.736)	0.398 (0.359, 0.434)	0.716 (0.681, 0.750)	0.521 (0.493, 0.548)
Recurrent neural network (without attention mechanism)	0.617 (0.586, 0.648)	0.327 (0.292, 0.363)	0.393 (0.353, 0.431)	0.275 (0.245, 0.304)
Recurrent neural network (with attention mechanism)	0.694 (0.666, 0.722)	0.468 (0.432, 0.502)	0.580 (0.541, 0.617)	0.474 (0.416, 0.536)
Hierarchical attention network (no pretraining, word attention only)	0.695 (0.666, 0.725)	0.405 (0.367, 0.443)	0.473 (0.437, 0.512)	0.341 (0.302, 0.390)
Hierarchical attention network (no pretraining, line attention only)	0.731 (0.704, 0.760)	0.464 (0.425, 0.503)	0.473 (0.434, 0.512)	0.340 (0.301, 0.388)
Hierarchical attention network (no pretraining, word and line attention)	0.784 (0.759, 0.810)	0.566 (0.525, 0.607)	**0.916 (0.895, 0.936)**	**0.841 (0.778,0 .895)**
Hierarchical attention network (with pretraining, word and line attention)	**0.800 (0.776, 0.825)**	**0.594 (0.553, 0.636)**	0.904 (0.881, 0.927)	0.822 (0.744, 0.883)

Classifier performance and confidence intervals on individual tasks are shown within parentheses.
The bolded values in the respective columns highlight the best performing classifier.

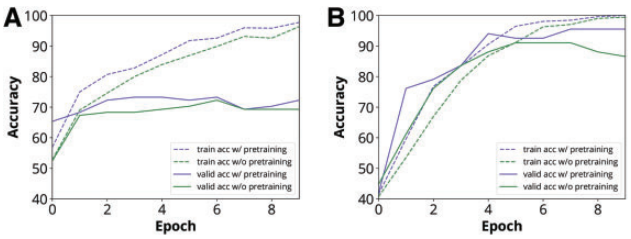

Figure 2. The HAN trains and validates accuracies with and without pretraining during the first 10 epochs for (**A**) the primary site classification task and (**B**) the histological grade classification task.

dimensionality reduction technique such as principal component analysis to plot the similarity of the pathology reports in our corpus based on their corresponding document embeddings (Figure 4). From this plot, we can see the natural clustering that occurs between pathology reports with similar cancers. We can also use this technique to locate and better understand misclassified reports based on the location of their document embeddings relative to existing document clusters. We note that the document embeddings generated by the HAN segments the different classes significantly better than unsupervised techniques such as TF-IDF–weighted word embeddings (Supplementary Figure S3) or Doc2Vec (Supplementary Figure S4). The HAN embeddings are also better segmented than document embeddings generated by the CNN (Supplementary Figure S5).

Error analysis

We analyzed the confusion matrices for the HAN in both the primary site and histological grade classification tasks (Figure 5). In the primary site classification task, we can see that the vast majority of misclassifications were within the same type of cancer rather than between the 2 types of cancer. In other words, misclassified breast cancer sites were mostly misidentified as other breast cancer sites rather than as lung cancer sites, and vice versa.

As expected, the sites with the highest numbers of samples had the highest individual *F*-scores, while the sites with the lowest numbers of samples had the lowest *F*-scores. The largest proportion of misclassification was related to the labels c34.9 and c50.9, which are used to label lung cancer reports where the subsite is not specified and breast cancer reports where the subsite is not specified, respectively. In these reports, the HAN may have mistakenly identified features suggesting a particular subsite. Another label with a high number of misclassifications was c50.8, which is used to identify reports for breast cancer that overlap several subsites. In these reports, the HAN may have identified only one of the subsites and failed to recognize that several subsites were involved. We note that the HAN had stronger performance dealing with the ambiguities in labels c34.9, c50.8, and c50.9 compared to the other classification models based on the confusion matrices for each model (Supplementary Figures S6–13).

In the histological grade classification task, G4 was the least populated, with only 17 samples, and therefore had the highest proportion of misclassifications. Among the other 3 grades, the HAN maintained strong performance.

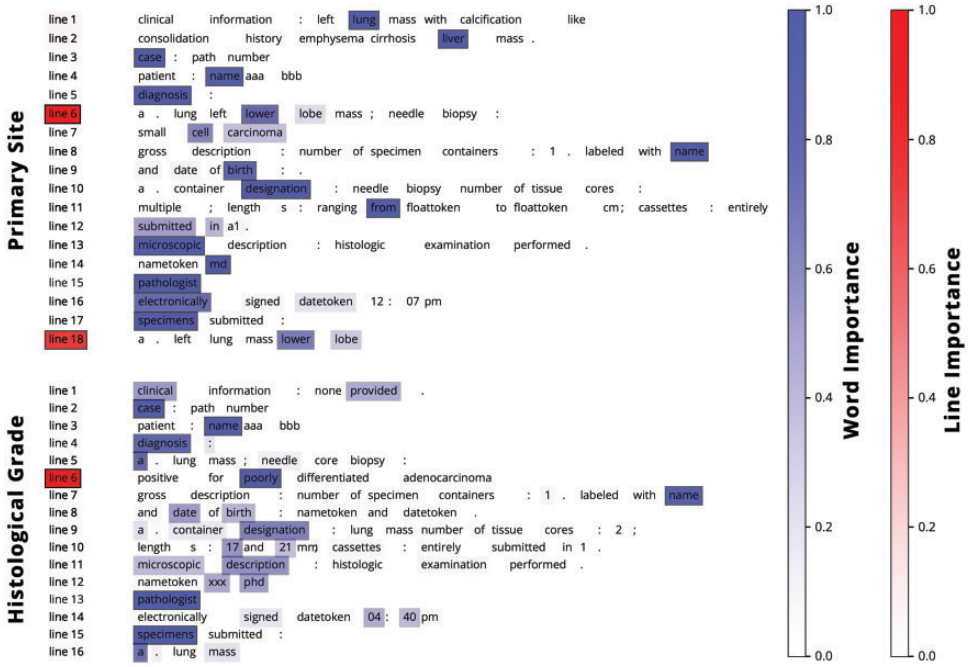

Figure 3. HAN annotations on sample pathology report for each classification task. The most important words in each line are highlighted in blue, with darker blue indicating higher importance. The most important lines in the report are highlighted in red, with darker red indicating higher importance. For each task, the HAN can successfully locate the specific line(s) within a document and text within the line(s) that identify the primary site (eg, lower lobe) or histological grade (eg, poorly differentiated). The RNN structure utilized by the HAN allows it to take into account word and line context to better locate the correct text segments.

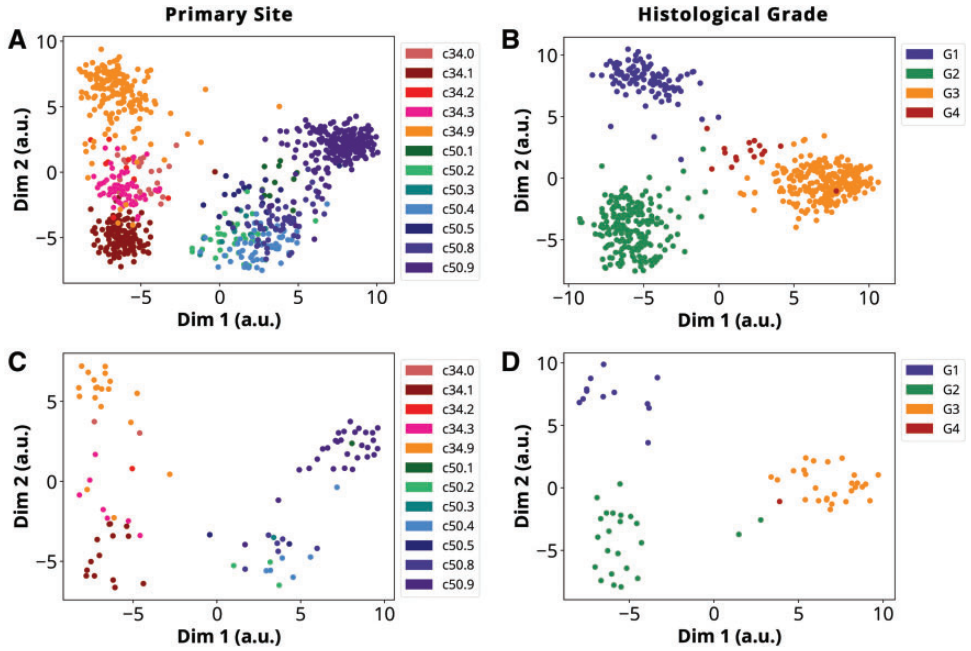

Figure 4. HAN document embeddings reduced to 2 dimensions via principal component analysis for (**A**) primary site train reports, (**B**) histological grade train reports, (**C**) primary site test reports, and (**D**) histological grade test reports.

DISCUSSION AND CONCLUSIONS

Based on the results in Table 2, we have shown that HANs can classify pathology reports better than traditional ML classifiers that use TF-IDF features. Because vector space models such as TF-IDF and bag-of-words cannot take into account the ordering of words beyond a small window, ML classifiers that rely on these representations can only make decisions based on the appearance of specific word tokens, irrespective of their context. Cancer pathology reports (and other unstructured text documents) typically include complex information spread over long segments of text, and our results suggest that vector space models are unable to effectively capture the deeper and more abstract linguistic patterns required for higher-level information abstraction.

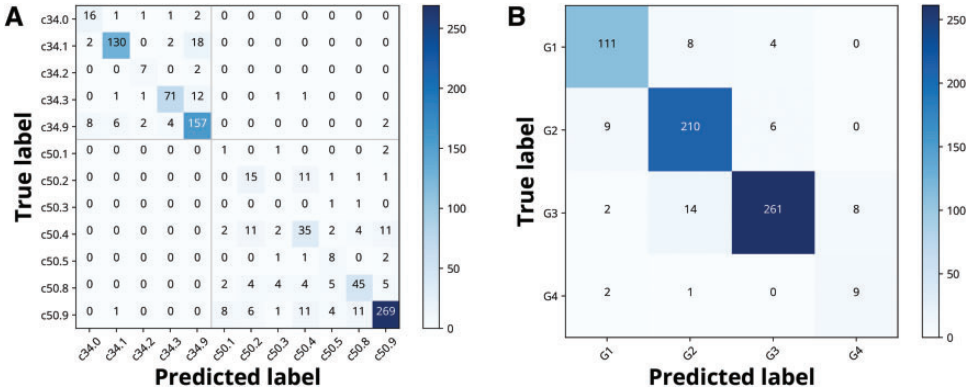

Figure 5. Confusion matrix for (**A**) HAN with pretraining on the primary site classification task and (**B**) HAN without pretraining on the histological grade classification task.

On the other hand, HANs do not suffer as much from this shortcoming, because their structure is designed to capture meaningful linguistic patterns across longer sequences of words within a sentence and sentences within a document. The use of RNNs within the HAN structure enables the HAN to distinguish nuances between the same words or word groups across different contexts. Furthermore, the HAN (and other DL models) rely on word embedding representations of words rather than the word tokens themselves; because these word embedding representations capture word similarity, this allows the HAN to better process never-before-seen word sequences based on their similarity to existing words and sequences. The high performance of the HAN suggests that the combination of these approaches is better at capturing important information than making decisions based simply on the appearance of specific words or groups of words.

Our results also show that HANs outperform other DL architectures. As in TF-IDF–based models, CNNs are limited in that they can only process a maximum window of 5 (in our implementation); any linguistic context beyond this window cannot be captured. RNN-based networks can process much longer word segments, allowing them to capture meaningful linguistic abstractions across longer contexts of words. However, basic RNNs have a shortcoming, in that they can have difficulty retaining important information over a very large number of timesteps.[25] While an attention mechanism mitigates this issue, our results show that attention mechanisms are not effective enough for long documents such as pathology reports.

HANs address these issues by taking into account the natural segmentation of text. The HAN first creates a representation of each sentence in a document based on the most important words in that sentence; it then creates a representation of the entire document based on the most important sentences in the document. By breaking down long documents into smaller chunks, the HAN can better locate and extract critical information. Our results suggest that this approach works more effectively than other artificial methods for processing text.

By examining the internal weights of the HAN, we can visualize task-specific words/lines that enable an understanding of how the pathology reports are being processed. These visualizations, apart from confirming that the HAN is learning the correct representations for a document, can enable clinicians to further analyze and annotate pathology reports for additional findings. In particular, the HAN can first identify the segments in a pathology report relevant to a specific classification task, and then a human expert can follow up and use the annotations from the HAN in other related tasks. We posit that this represents a first step toward automated annotation of clinical texts.

Pretraining the HAN provides an additional performance boost for the primary site classification task by allowing the HAN to first learn the inherent structure of a corpus before further tuning the weights on a fine-grained classification task (Figure 2). This is especially useful because many pathology reports available in cancer registries are unlabeled. Our pretraining approach allows the HAN (and other similar DL models) to benefit from these reports without requiring a trained expert to manually create labels first.[22] A natural extension of the pretraining is to use these learned features as priors for semisupervised models.[26,27]

A study by Powsner and colleagues[28] showed that surgeons misunderstand pathology reports up to 30% of the time. With our method, we aim to reduce both the time required and the error rate for pathology report classification. In the primary site classification task, our HAN achieved a micro *F*-score performance level of 0.80. We believe that performance could be improved even further if not for the inherent ambiguity in the labels for 3 classes, c34.9, c50.8, and c50.9, which represent documents that were not annotated with a subsite or were annotated with multiple subsites. In the histological grade classification task, where this ambiguity did not exist, the HAN achieved a micro *F*-score performance level of 0.92. Furthermore, the HAN's performance was limited by the relatively small size of our dataset; for several classes we had only tens of samples for the HAN to train on. We believe that, if provided with a larger dataset, the HAN could better learn the linguistic patterns in the reports and achieve a level of performance that would surpass that of human experts, enabling the HAN to play an effective and practical role in the automation of cancer pathology report classification.

Due to the complexity of the HAN architecture relative to other ML architectures, the HAN takes longer to process documents compared to the other models tested. However, this limitation does not preclude practical application of the HAN for document classification. On a single central processing unit, the HAN can still process an entire corpus of 2505 documents in ~150 s (Supplementary Table S3). We expect that with the emergence of newer computer architectures and better training approaches for RNNs, both the training and testing time performance can be significantly improved. Analyses of scaling and performance behaviors of our implementation are currently under way.

One important limitation of our study is the small size of our dataset. Because DL models learn their own features from raw data,

they often require large datasets containing thousands or millions of samples to reach full effectiveness. Since our dataset used for supervised training consisted of <1000 samples, our DL approached risk overfitting on the data. Given a larger dataset, we expect the effectiveness of the HAN relative to other approaches to improve even further. Yang et al.[9] demonstrated that HANs significantly outperform CNNs, RNNs, and other traditional ML classifiers on large public datasets such as Yelp and IMDb, and we expect the same to hold true for our pathology reports and other biomedical datasets. To confirm this, we tested the performance of the HAN on a simple 8-class topic classification task with PubMed abstracts. We found that increasing the number of abstracts from 8000 to 80 000 to 800 000 yielded overall classification accuracies of 70.63%, 76.50%, and 76.88%; additional details are available on our supporting information page, Supplementary 10–11. We also have plans to run our pathology report classification experiments on a much larger dataset of 20 000 labeled pathology reports that was unavailable during the time of this study.

Moving forward, we will extend the HAN to perform simultaneous classification on multiple tasks. DL models adapted for multitask learning have been shown to have higher performance on subtasks compared to the same models adapted for single-task classification.[29] By adapting the HAN to simultaneously predict both primary site and histological grade, we expect to be able to increase performance on both tasks. Another promising area of research lies in automating the generation of pathology reports and other medical documents from raw imaging data or other raw medical data sources. Like the decoding process for a cancer pathology report, the encoding process of tumor data into a cancer pathology report is subject to human error and variability in the interpretation of coding guidelines. An effective automated solution could standardize how tumor data are encoded into pathology reports and yield improvements in the accuracy and efficiency of the report-processing pipeline.

The Python code developed based on TensorFlow is available at: http://code.ornl.gov/v33/PathRepHAN. Enhancements for multiple graphics processing unit–based training and performance/scaling behaviors will be documented as part of further development.

COMPETING INTERESTS

The authors have no competing interests to declare.

FUNDING

This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer program established by the US Department of Energy and the National Cancer Institute of the National Institutes of Health. This work was performed under the auspices of the US Department of Energy by Argonne National Laboratory under contract DE-AC02-06-CH11357, Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344, Los Alamos National Laboratory under contract DE-AC5206NA25396, and Oak Ridge National Laboratory under contract DE-AC05-00OR22725. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the US Department of Energy Office of Science and the National Nuclear Security Administration. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the US Department of Energy under contract DE-AC05- 00OR22725.

CONTRIBUTORS

SG implemented, tested, and validated the experiments. All authors were involved in designing and developing the study and writing the paper.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

ACKNOWLEDGMENTS

The authors wish to thank Valentina Petkov of the NCI Surveillance Research Program and the SEER registries in Hawaii, Kentucky, Connecticut, New Mexico, and Seattle, Washington, for the deidentified pathology reports used in this investigation.

REFERENCES

1. Lowy D, Collins F. Aiming high—changing the trajectory for cancer. *New Engl J Med*. 2016;374(20):1901–04.
2. National Cancer Institute. *Overview of the SEER Program*. 2017. https://seer.cancer.gov/about/overview.html. Accessed October 10, 2017.
3. Kumar A, Irsoy O, Ondruska P, *et al*. Ask me anything: dynamic memory networks for natural language processing. In: *Proc Int Conf Mach Learn*. 2016:1378–87.
4. Kim Y. Convolutional neural networks for sentence classification. arXiv preprint arXiv:14085882. 2014.
5. Lipton Z, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:150600019. 2015.
6. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735–80.
7. Chung J, Gulcehre C, Cho K, *et al*. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:14123555. 2014.
8. Graves A. Generating sequences with recurrent neural networks. arXiv preprint arXiv:13080850. 2013.
9. Yang Z, Yang D, Dyer C, *et al*. Hierarchical attention networks for document classification. *In:* Proceedings of NAACL-HLT. 2016: 1480–89.
10. Carrell D, Halgrim S, Tran D, *et al*. Using natural language processing to improve efficiency of manual chart abstraction in research: the case of breast cancer recurrence. *Am J Epidemiol*. 2014;179(6):749–58.
11. Martinez D, Li Y. Information extraction from pathology reports in a hospital setting. In: *Proc ACM Int Conf Inf Knowl Manag*. ACM; 2011:1877–82.
12. Li P, Huang H. Clinical information extraction via convolutional neural network. arXiv preprint arXiv:160309381. 2016.
13. Jouhet V, Defossez G, Burgun A, *et al*. Automated classification of free-text pathology reports for registration of incident cases of cancer. Methods Inf Med. 2012;51(3):242.
14. Jagannatha A, Yu H. Bidirectional RNN for medical event detection in electronic health records. In: Proceedings of NAACL-HLT. NIH Public Access; 2016:473.
15. Qiu J, Yoon H, Fearn P, *et al*. Deep learning for automated extraction of primary sites from cancer pathology reports. *IEEE J Biomed Health Inform*. 2017. https://doi.org/10.1109/JBHI.2017.2700722.
16. National Cancer Institute. *Coding Guidelines Breast C500–C509*. 2016. https://seer.cancer.gov/ archive/manuals/2010/AppendixC/breast/coding_guidelines.pdf. Accessed July 15, 2017.
17. Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. In: *Proc 26th Intl Conf Neural Inf Process Syst*. 2013;2:3111–19.
18. Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.
19. Pennington J, Socher R, Manning C. GloVe: global vectors for word representation. In: Proc Conf Empir Methods Nat Lang Process. 2014;14:1532–43.
20. Greff K, Srivastava R, Koutník J, *et al*. LSTM: A search space odyssey. *IEEE Trans Neural Netw Learn Syst*. 2016.
21. Bernstein B, Potts C. *Optimizing the Hyperparameter of Which Hyperparameter Optimizer to Use*. 2017. https://roamanalytics.com/2016/09/15/optimizing-the-hyperparameter-of-which-hyperparameter-optimizer-to-use/. Accessed July 15, 2017.

22. Erhan D, Bengio Y, Courville A, *et al*. Why does unsupervised pre-training help deep learning? *J Mach Learn Res*. 2010;11:625–60.
23. Li J, Luong ML, Jurafsky D. A hierarchical neural autoencoder for paragraphs and documents. In: *Proc 53rd Annu Mtg Assoc Comput Linguist*. 2015: 1106–15.
24. DiCiccio T, Efron B. Bootstrap confidence intervals. *Stat Sci*. 1996;11(3):189–212.
25. Chorowski J, Bahdanau D, Serdyuk D, *et al*. Attention-based models for speech recognition. *In:* Adv Neural Inf Process Syst. 2015: 577–85.
26. Johnson R, Zhang T. Semi-supervised convolutional neural networks for text categorization via region embedding. In: Adv Neural Inf Process Syst *NIPS '15*. Cambridge, MA: MIT Press; 2015: 919–27.
27. Johnson R, Zhang T. Supervised and semi-supervised text categorization using LSTM for region embeddings. In: Proc Int Conf Mach Learn. ICML '16. JMLR.org. 2016: 526–34.
28. Powsner S, Costa J, Homer R. Clinicians are from Mars and pathologists are from Venus: clinician interpretation of pathology reports. *Arch Pathol Lab Med*. 2000;124(7):1040–46.
29. Yoon H, Ramanathan A, Tourassi G. Multi-task deep neural networks for automated extraction of primary site and laterality information from cancer pathology reports. In: *Advances in Big Data: Proceedings of the INNS Conference on Big Data*. Cham, Switzerland: Springer; 2016: 195–204.