Risk Identification for Plant Health via Text Mining of E-Commerce Data

Yanran Guan
Canadian Food Inspection Agency
yanran.guan@canada.ca

August 8, 2019

Abstract

E-commerce is a source of risk to plant health as it can act as a pathway to the selling and purchasing of harmful organisms. In this work, using the textual product information collected from the e-commerce platform *Alibaba*, we conduct a comparative study through training a series of predictive text mining models to identify the risks of concern to plant health from on-line traded products. The experimental results show that, among all the trained models, the FastText model achieves the highest F_1-score of 92.31% on the test data. A web application of an automated risk identification tool is thereby developed based on the FastText model to improve the risk management with respect to plant health.

1 Introduction

Plant health is vital to the protection of eco-environment and to the sustainable development of bio-economy. Managing risks to Canada's plant resources, which includes regulating imports of plants and plant-related products, is one of the responsibilities of the Canadian Food Inspection Agency (CFIA). Today, the on-line trade provided by e-commerce platforms plays an important role in a country's importation and exportation of goods, which also provides a channel for some non-compliant goods entering the country without detection. Therefore, the objective of this research is to develop a predictive analytics method that identifies the on-line traded products that pose risks to plant health in Canada through automated data collection, which will further allow CFIA to focus efforts on the e-traders of *high-risk* articles, including working with them to better inform purchasers before shipment.

Since the product information is largely presented in textual data such as product titles and product descriptions, in this work, we adopt predictive text mining algorithms for risk identification of on-line traded products. We first apply the predictive text mining model FastText [1] to the textual data to learn the word and text vectors [2] and perform text classification [3]. Next, we reuse the text vectors derived by FastText as the input layer of other predictive models, including k-nearest neighbors (k-NN) [4], naïve Bayes [5], classification and regression tree (CART) [6], C5.0 [7], random forest [8], extreme gradient boosting (XGBoost) [9], and support vector machine (SVM) [10].

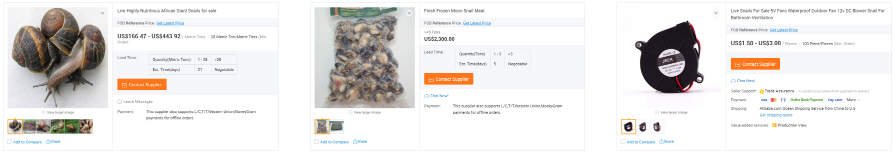
The trained models are evaluated based on the accuracy, precision, recall, and F_1-score of their predictions of risk levels. The optimal model is selected to be integrated with a web crawler based search engine which captures product information from e-commerce platforms using search terms and to be fed with the textual data collected by the search engine. To facilitate user access, we deploy the model and the search engine into a web application, which can automatically provide the user with an identified list of *high-risk* and *low-risk* product entries according to the search term entered by the user.

1.1 Risks to Plant Health of E-Commerce

With the development of e-commerce, the trade of commodities between sellers and buyers from different countries has been largely facilitated, which also increases the risks to plant health. Potentially harmful organisms such as invasive plants, seeds and insects, as well as commodities or articles capable of carrying or spreading these pests, can easily be offered for sale and purchased from anywhere in the world. In recent decades, e-commerce has become a novel yet growing distribution channel of horticultural products [11] and is recognized as a particularly important driver of biological invasions [12]. Researchers found that, in 2015, over 500 invasive species were traded worldwide on e-commerce websites every day, and the proportion of invasive species available on-line was even higher than that of non-invasive species [12]. In New Zealand, the importation of unwanted flora and fauna is considered as a biosecurity risk associated with on-line trading [13]. It is also documented that e-commerce, being the major mode of dispersal for many species of *Caulerpa*, poses a high risk of the invasion of the aquarium strain in the Mediterranean, southern California, and Australia [14].

1.2 Text Mining in Risk Management

Risk management consists in the identification and evaluation of risks, together with the coordinated applications that avoid or minimize the impact of risks. In the last decades, diverse data mining and machine learning approaches have been introduced to the field of risk management that perform risk analysis and prediction. In this context, unstructured data, i.e., textual data, is a large source of information that data mining and machine learning algorithms can utilize. Previous empirical research has shown that, in financial risk management, analysis and modeling based on textual data can improve the prediction of the impact of corporate and regulatory disclosures to intraday stock prices [15] and to long-term stock index [16]. In terms of managing safety risks, predictive and descriptive models applied on unstructured accident datasets such as accident reports were developed for risk assessment of occupational accidents in the steel



Live Highly Nutritious African Giant Snails for sale

FOB Reference Price: Get Latest Price

US$166.47 - US$443.92 / Metric Tons | 28 Metric Ton/Metric Tons (Min. Order)

Lead Time:

Quantity(Metric Tons)	1 - 28	>28
Est. Time(days)	21	Negotiable

Payment: This supplier also supports L/C,T/T,Western Union,MoneyGram payments for offline orders.

(a)

Fresh Frozen Moon Snail Meat

FOB Reference Price: Get Latest Price

>=5 Tons
US$2,300.00

Lead Time:

Quantity(Tons)	1 - 3	>3
Est. Time(days)	5	Negotiable

Payment: This supplier also supports L/C,T/T,Western Union,MoneyGram payments for offline orders.

(b)

Live Snails For Sale 5V Fans Waterproof Outdoor Fan 12v DC Blower Snail For Bathroom Ventilation

FOB Reference Price: Get Latest Price

US$1.50 - US$3.00 / Pieces | 100 Piece/Pieces (Min. Order)

Seller Support: Trade Assurance – To protect your orders from payment to delivery

Payment: VISA TT Online Bank Payment Pay Later More

Shipping: Alibaba.com Ocean Shipping Service from China to U.S
Get shipping quote

Value-added services: Production View

(c)

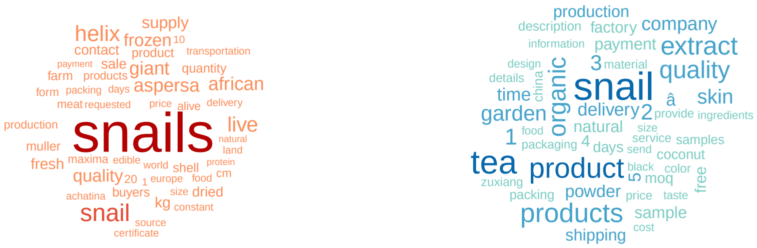
Figure 1: Examples of products captured by the search term *live snail*, among which (a) is labeled as *high-risk*, whereas (b) and (c) are labeled as *low-risk*, as only living snails are considered non-compliant from the perspective of plant protection [25].

industry [17] and for identification of risk factors in the construction of urban rail transit [18]. Software project managers also benefit from text mining: in the research of Huang et al. [19], an automated approach was proposed based on text classifiers to detect technical debt in project code for the maintenance of long term software projects; according to the work of Vijayakumar et al. [20], the risks during the software development process in cloud environments, such as privacy and security flaws, can be detected in real-time through contextual and semantic text analysis when committing the code. Besides, in the recent studies of bio-informatics, text mining approaches have been widely used to identify risk factors of multiple diseases [21, 22, 23, 24].

2 Data Preparation and Data Profiling

In this work, the keyword *snail*, which refers to a category of products subject to the import requirements under the Canadian *Plant Protection Act* [25], is selected as the object of our case study, and the on-line trading website *Alibaba* as the target e-commerce platform of data acquisition. To capture as much data as possible, the experimental data is collected from *Alibaba* using 6 search terms synonymous to *snail*. They are respectively, *achatina*, *brown garden snail*, *edible snail*, *giant African snail*, *helix snail*, and *live snail*.

Using the aforementioned search terms, a dataset comprised of 1040 different instances of product advertisement is collected from the *Alibaba* website. Each instance consists of 3 attributes: (i) *url*, the URL link to the product home page; (ii) *title*, a short description within 60 letters of the concerned product, which is mandatory for all product sellers to provide; (iii) *description*, a detailed description with no text length limitation, but is optional and can be missing for a few instances. The values of the textual attributes of an instance, i.e., *title* and *description*, are encoded in UTF-8. Moreover, under the guidance of the *Plant Protection Act* and according to the information provided on the product pages, we add to each instance manually an additional binary attribute *risk*, of which the value is either *high-risk* (the positive class) or *low-risk* (the negative class), indicating the risk level of the concerned product. To better illustrate



(a) (b)

Figure 2: Word clouds of the top 50 common words from instances grouped by (a) *high-risk* and (b) *low-risk*.

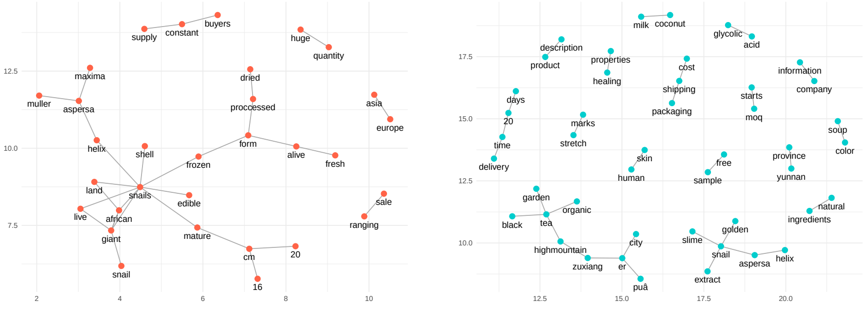
the scheme of risk labeling, a few examples of collected products are given in Figure 1. Following this labeling scheme, 384 instances are labeled as *high-risk* and 656 instances are *low-risk*.

In order to make full use of the textual information in the product advertisements, we concatenate the texts of the *title* and *description* attributes for each product instance to prepare the experimental text. We first tokenize the text instances by single words. After removing stop words, the common words from instances grouped by the two risk levels are visualized in the form of word clouds in Figure 2. An interesting pattern can be found in the word clouds is that the most frequent *high-risk* word is *snails*, while the most frequent *low-risk* word is *snail*. The plural and singular forms of the same word indicate different word usages: *snail* can be used as a descriptive word to characterize the properties of a product, e.g., the shape; *snails* occurs mostly in instances concerning the real animal.

The relationship between words is another interesting direction to explore, so we tokenize the texts into pairs of two consecutive words, called bigrams. After filtering out the bigrams containing stop words, we visualize the most common bigrams in Figure 3 in the form of grammar graphs from instances grouped by the two risk levels. We could find from Figure 3 a few patterns revealing the structure of the experimental text instances. The largest subgraph in Figure 3(a) is formed around the central node of the word *snails*. In Figure 3(b), there are two major subgraphs: the bottom left one formed around *tea* describes the tea-related products; the bottom right one around *snail* describes the by-products of snail, such as snail slime and snail extract.

3 Predictive Modeling

In this section, we briefly introduce the predictive modeling methods used in our experiment, describe the parameter settings for each modeling algorithm, and present the test results of the trained models. Prior to building the models, stratified sampling is applied to the dataset using the target attribute *risk* as



(a) (b)

Figure 3: Grammar graphs of the top 30 common bigrams from instances grouped by (a) *high-risk* and (b) *low-risk*.

the stratification variable, which partitions the dataset into an 80%/20% split for training and testing.

3.1 Modeling Methods

In order to find the best model that works with our data, we experiment with multiple predictive modeling algorithms. We first apply the FastText model to our data, and then reuse the text vectors that are learned in the training process of FastText for other predictive models. All models are trained on the training set and evaluated on the test set.

3.1.1 FastText

The FastText model is comprised of two parts: (i) the text classification part, which is a supervised learner that takes word vectors as input and predicted classes as output. In this experiment, the FastText model is trained on the training set and tested on the test set; (ii) the word embedding part, called Word2Vec [2], which is an unsupervised learner that maps for each n-gram token its one-hot encoding into a low-dimensional vector based on the surrounding tokens.

Text classifier. The structure of the text classifier of FastText is illustrated in Figure 4. In the input layer, the N n-grams of each text instance are represented in the form of h-dimensional word vector $\mathbf{s}_i$ derived from Word2Vec, where $i \in \{1, 2, \ldots, N\}$. Next, the classifier computes the text vector $\mathbf{x} = (x_1, x_2, \ldots, x_h)$ by averaging all the word vectors $\mathbf{s}_i$ of the text. The text vector is then fed to the hidden layer and multiplied by an h by m weight matrix $\mathbf{M}_{h\times m}$ which transforms the text vector to an m-dimensional classification vector $\mathbf{z} = (z_1, z_2, \ldots, z_m)$, where m is the number of text classes. In the output layer, the classification

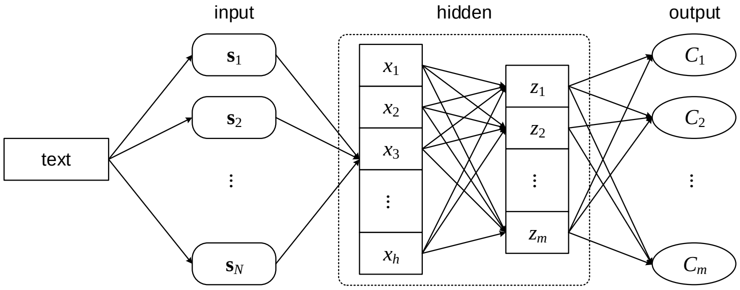


Figure 4: The text classifier of FastText.

for each class C_i, where $i \in \{1, 2, \ldots, m\}$, is represented by the probability computed by the softmax function, written as:

$$P(C_i) = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}}. \tag{1}$$

In this experiment, the texts are classified into two classes, i.e. *high-risk* and *low-risk*. The number of text classes m is set as 2.

Word2Vec. There are two types of Word2Vec models to compute the word vectors, namely the continuous bag-of-words model (CBoW) and the continuous skip-gram model (skip-gram). The diagrammatic representations of the two models are illustrated in Figure 5, using the window size of 5 of the surrounding tokens as an example.

For both CBoW and skip-gram, the input layer and output layer are comprised of v-dimensional one-hot encoded vectors of n-gram tokens. Between the input and output layers, there are two sets of weights composing two matrices: the matrix $\mathbf{W}_{v\times h}$ that transforms the v-dimensional input space to the h-dimensional hidden space $\mathbf{s} = (s_1, s_2, \ldots, s_h)$, and the transposed matrix $\mathbf{W}'_{h\times v}$ of $\mathbf{W}_{v\times h}$ that maps the h-dimensional hidden space to the v-dimensional output space. Similar to the text classifier, the softmax function is applied to the end of the output layer, so that each element of the output layer describes the likelihood of a specific token that will appear in the context. The difference between CBoW and skip-gram lies in the position where the target token $\mathbf{w}_p$ is placed in the model. CBoW uses the one-hot encoding of target token as input and the average of one-hot encodings of the tokens surrounding the target one as output, while skip-gram uses the one-hot encoding of target token as output and the averaged encoding of the surrounding ones as input. Through training the Word2Vec models, the weight matrices $\mathbf{W}_{v\times h}$ and $\mathbf{W}'_{h\times v}$ are getting optimized and the word vector of each target token is obtained by extracting the h-dimensional hidden layer.

In this experiment, the experimental texts are tokenized into bigrams. We use the skip-gram model to learn the word vectors of the bigrams and set the

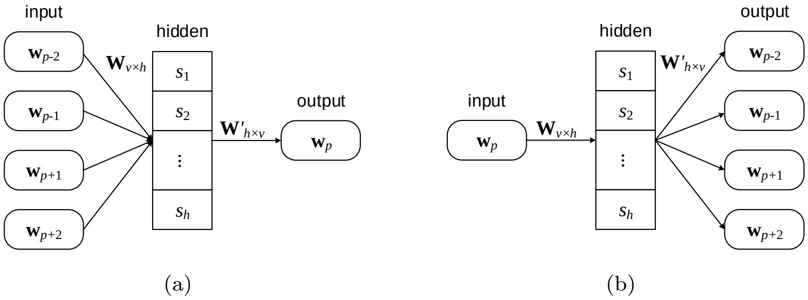


Figure 5: The Word2Vec models: (a) CBoW and (b) skip-gram.

size h of the hidden layer to 20.

3.1.2 Reuse of Text Vectors

After the FastText model has been trained, we use the averaged word vector as the text vector of each text instance to train some other predictive models, such as k-NN, naïve Bayes, CART, C5.0, random forest, XGBoost, and SVM. Each model is trained with 10-fold cross-validation. For the models with parameters to be tuned, we try for each model 10 different combinations of parameters with a tuning objective of maximizing the area under curve (AUC) of the receiver operating characteristic (ROC) curve.

k-NN. The k-NN algorithm is an instance-based learning algorithm that does not require specific training except for adjusting the heuristic value k. Given a set of pairs of text vectors and their corresponding class labels, k-NN classifies an instance according to the majority class of its k closest neighbor instances in the feature space. In this experiment, k is tuned to 23.

Naïve Bayes. Naïve Bayes is a conditional probability classifier based on Bayes' theorem. It assumes that all the features used for classification are independent of each other. Specifically, for a test instance $\mathbf{x} = (x_1, x_2, \ldots, x_{20})$ represented by a 20-dimensional text vector, its classification $\hat{y}$ is given by naïve Bayes based on the following equation:

$$\hat{y} = \arg\max_{c} P(c) \prod_{i=1}^{20} P(x_i|c), \tag{2}$$

where c refers to the class to be classified, which corresponds to the two risk levels, i.e., $c \in \{C_1, C_2\}$. $P(c)$ is the occurrence probability of class c in the training set, and $P(x_i|c)$, where $i \in \{1, 2, \ldots, 20\}$, is the conditional occurrence probability of the feature x_i given class c. Since all features in the embedding space are numeric, we use the probability density values derived from kernel density estimation (KDE) [26] of the numeric features as their conditional probabilities.

CART. CART is a binary decision tree algorithm that partitions the text embedding space into disjoint regions represented by terminal nodes. It has been used and explained in our previous project [27]. CART uses Gini index as the splitting criterion and uses the complexity cost value cc to control the splitting size of the tree, which is calculated as:

$$cc = \sum_{i=1}^{n_{\mathrm{tm}}} \mu_i + \lambda n_{\mathrm{split}}, \tag{3}$$

where n_{tm} is the number of terminal nodes of a given tree, n_{split} is the total split number of the tree, μ_i, where $i \in \{1, 2, \ldots, n_{\mathrm{tm}}\}$, is the number of misclassification on the i^{th} terminal node, and λ is a constant penalty term determined through cross-validation. The splitting of CART stops when its cc is reduced to less then a threshold value. In this experiment, the optimal CART model is found with a threshold of cc of 0.8571.

C5.0. C5.0 is another decision tree based algorithm. It is extended from C4.5 [7] and offers more features than C4.5, such as winnowing and boosting. Different from CART, C5.0 can generate both binary tree and multi-branch tree. C5.0 uses information gain as the splitting criterion and uses the binomial confidence limit method as the post-pruning method to control the tree size. We train the C5.0 model with adaptive boosting (AdaBoost) [28] and the best model is boosted with 80 trials.

Random forest. Random forest is an ensemble learning algorithm that trains a multitude of decision trees and makes classification using the mode of the predicted classes of the individual trees. We use the extremely randomized trees (Extra-Trees) [29] as the implementation of random forest. In Extra-Trees, at each tree node of the individual trees, the split is made upon a random choice of a certain number of candidate features, which is tuned to 12 in our model.

XGBoost. XGBoost is a framework of gradient boosted models. For our model, we choose CART as its base learner. XGBoost sums the regression scores on each leaf node of the individual CART models and uses the summed score as the final prediction. The CART models are generated iteratively in XGBoost and each CART is built upon a subsampled set of features of subsampled training instances. In this experiment, the subsampling rate of instances is tuned to 0.89 and the subsampling rate of features is 0.80. Besides, the number of boosting iterations is set to 300.

SVM. SVM is a kind of non-probabilistic binary classifier. It uses linear or non-linear kernel functions to map the input instances into a high-dimensional feature space and separate the instances from different classes in the high-dimensional space with a hyperplane that divides them by a clear gap as wide as possible. We adopt three types of kernel functions, respectively, linear function, polynomial function and radial basis function (RBF). Given two text vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, where $i \neq j$, the kernel functions $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ are written as:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \mathbf{x}_i \cdot \mathbf{x}_j, & \text{linear kernel,} \\ (\gamma \mathbf{x}_i \cdot \mathbf{x}_j + b)^d, & \text{polynomial kernel,} \\ e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, & \text{RBF kernel,} \end{cases} \tag{4}$$

where the coefficients γ, b, and d are needed to be specified for each corresponding kernel function. In this experiment, for the polynomial kernel SVM, the three coefficients are tuned as: $\gamma = 0.1$, $b = 0.5$, and $d = 3$, and for the RBF kernel SVM, the coefficient γ is tuned as 0.39.

3.2 Test Results

To compare the performance between the different models and find the best one for risk identification, we test the trained models on the test set and evaluate their predictive results. As summarized in Table 1, a comprehensive analysis is made on the test results of the models from the following aspects: accuracy, precision, recall, and F_1-score. The results show that the FastText model, which achieves an F_1-score of 92.31% and an accuracy of 94.20%, is dramatically superior to the other models in terms of identifying *high-risk* instances in the test data.

Table 1: Comparative analysis of the models.

Model	Accuracy	Precision	Recall	F_1-score
FastText ($h = 20$, bigram)	94.20%	90.00%	94.74%	92.31%
Skip-gram + k-NN	89.37%	86.49%	84.21%	85.33%
Skip-gram + naïve Bayes	91.30%	91.43%	84.21%	87.67%
Skip-gram + CART	90.82%	88.00%	86.84%	87.42%
Skip-gram + C5.0	90.34%	87.84%	85.53%	86.67%
Skip-gram + random forest	89.86%	85.71%	86.84%	86.27%
Skip-gram + XGBoost	90.34%	86.84%	86.84%	86.84%
Skip-gram + linear SVM	89.86%	91.04%	80.26%	85.31%
Skip-gram + polynomial SVM	90.34%	91.18%	81.58%	86.11%
Skip-gram + RBF SVM	89.86%	89.86%	90.79%	86.79%

Another comparative analysis is made of the different models based on the ROC curve, which is a commonly used graphical representation of the performance of binary classifications. The horizontal axis of the ROC curve represents the false positive rate (FPR), which is the fraction of false positives out of total actual negatives, and the vertical axis represents the true positive rate (TPR),

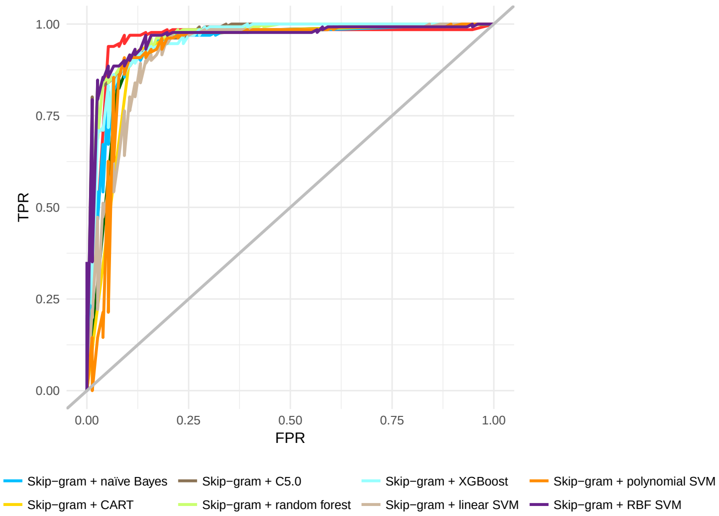


Figure 6: The ROC curves.

which is the fraction of true positives out of total actual positives. The ROC curves of the predictions on the test set of the models are shown in Figure 6, from which it can also be concluded that, the FastText model, represented by the red curve, having the highest AUC of 95.51% among all the models, prevails over the other models.

4 Model Deployment

Based on the optimal trained model, i.e., the FastText model, the web application, the plant health automated e-commerce data extractor (PHAEDE)[1], is developed that can automatically collect product information from *Alibaba* using user-entered search terms and perform risk identification for the collected data in real-time. The flow diagram in Figure 7 illustrates the operation flow implemented in PHAEDE.

The search keyword entered by the user, such as *live snail*, is first encapsulated into HTTP GET requests sent to the *Alibaba* website. With the help of the `jQuery.get()` function of Ajax, the GET requests are sent to each product page in the search results of *Alibaba* in an asynchronous manner, which efficiently return a list of raw HTML data that contains the entire web content of the requested pages. Then, a set of `jQuery.find()` functions and regular expressions are used to parse the HTML data so as to retrieve the page URL, product title, and product description within the web pages. The retrieved

[1] The web application is available at `https://irmmodelling.shinyapps.io/phaede/` and the code is available at `https://github.com/cfia-data-science/PHAEDE`.

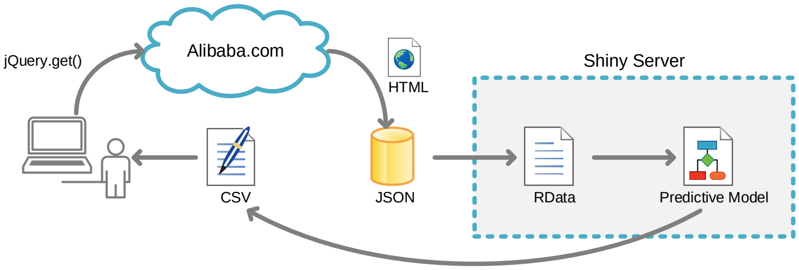


Figure 7: The workflow of PHAEDE.

data is stored in the structured data format JSON and converted into an R data frame which can be interpreted by the predictive model.

On the back-end side, since the predictive modeling experiment is conducted in R language, we use the Shiny Server of R to host the trained model. Once the data is collected and converted, the predictive model will be applied to the R data frame of the collected data and provide a list of predictive results. The R data frame will then be updated with concatenating each prediction into its corresponding instance and be converted into a CSV file sent back to the user for download.

5 Conclusion

In this work, we present the idea of using predictive text modeling algorithms to help identify the products with risks to plant health in e-commerce platforms. A comparative study is conducted to analyze the performance of different experimental models. Based on the results of the comparative study, the FastText model, which achieves the highest accuracy and F_1-score, is selected to be deployed into the web application PHAEDE, which can perform risk identification for on-line traded products through automated data collection and can improve the risk management regarding plant health of CFIA.

Further research can be made on more objects of case study, such as *earthworm*, *plant seeds*, and *soil*, which can also be potentially *high-risk* from the perspective of plant protection. Another interesting study is to use unsupervised learning to find the patterns in the textual data without human intervention. Specifically, different unsupervised learning models, such as clustering models, can be applied to the text embedding space derived by FastText to model the underlying structure of the data.

References

[1] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," *CoRR*,

vol. abs/1612.03651, 2016.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[3] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, (Valencia, Spain), pp. 427–431, Association for Computational Linguistics, Apr. 2017.

[4] T. M. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[5] D. J. Hand and K. Yu, "Idiot's Bayes—not so stupid after all?," *International Statistical Review*, vol. 69, no. 3, pp. 385–398, 2001.

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA, USA: Wadsworth and Brooks, 1984.

[7] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1993.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (San Francisco, CA, USA), pp. 785–794, ACM, Aug. 2016.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] K. Dehnen-Schmutz, O. Holdenrieder, M. J. Jeger, and M. Pautasso, "Structural change in the international horticultural industry: some implications for plant health," *Scientia Horticulturae*, vol. 125, no. 1, pp. 1–15, 2010.

[12] F. Humair, L. Humair, F. Kuhn, and C. Kueffer, "E-commerce trade in invasive plants," *Conservation Biology*, vol. 29, no. 6, pp. 1658–1665, 2015.

[13] J. G. Derraik and S. Phillips, "Online trade poses a threat to biosecurity in New Zealand," *Biological invasions*, vol. 12, no. 6, pp. 1477–1480, 2010.

[14] L. J. Walters, K. R. Brown, W. T. Stam, and J. L. Olsen, "E-commerce and Caulerpa: Unregulated dispersal of invasive species," *Frontiers in Ecology and the Environment*, vol. 4, no. 2, pp. 75–79, 2006.

[15] S. S. Groth and J. Muntermann, "An intraday market risk management approach based on textual analysis," *Decision Support Systems*, vol. 50, no. 4, pp. 680–691, 2011.

[16] S. Feuerriegel and J. Gordon, "Long-term stock index forecasting based on text mining of regulatory disclosures," *Decision Support Systems*, vol. 112, pp. 88–97, 2018.

[17] S. Sarkar, S. Vinay, and J. Maiti, "Text mining based safety risk assessment and prediction of occupational accidents in a steel plant," in *Proceedings of the 2016 International Conference on Computational Techniques in Information and Communication Technologies*, (New Delhi, India), pp. 439–444, IEEE, Mar. 2016.

[18] J. Li, J. Wang, N. Xu, Y. Hu, and C. Cui, "Importance degree research of safety risk management processes of urban rail transit based on text mining method," *Information*, vol. 9, no. 2, p. 26, 2018.

[19] Q. Huang, E. Shihab, X. Xia, D. Lo, and S. Li, "Identifying self-admitted technical debt in open source projects using text mining," *Empirical Software Engineering*, vol. 23, no. 1, pp. 418–451, 2018.

[20] K. Vijayakumar and C. Arun, "Automated risk identification using NLP in cloud based development environments," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2017.

[21] J. Urbain, "Mining heart disease risk factors in clinical text with named entity recognition and distributional semantic models," *Journal of Biomedical Informatics*, vol. 58, pp. S143–S149, 2015.

[22] J. Jonnagaddala, S.-T. Liaw, P. Ray, M. Kumar, N.-W. Chang, and H.-J. Dai, "Coronary artery disease risk assessment from unstructured electronic health records using text mining," *Journal of Biomedical Informatics*, vol. 58, pp. S203–S210, 2015.

[23] A. M. Small, D. H. Kiss, Y. Zlatsin, D. L. Birtwell, H. Williams, M. A. Guerraty, Y. Han, S. Anwaruddin, J. H. Holmes, J. A. Chirinos, *et al.*, "Text mining applied to electronic cardiovascular procedure reports to identify patients with trileaflet aortic stenosis and coronary artery disease," *Journal of Biomedical Informatics*, vol. 72, pp. 77–84, 2017.

[24] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Data processing and text mining technologies on electronic medical records: A review," *Journal of Healthcare Engineering*, vol. 2018, 2018.

[25] "Plant Protection Act." S.C. 1990, c. 22.

[26] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the 11th conference on Uncertainty in artificial intelligence*, (Montréal, QC, Canada), pp. 338–345, Morgan Kaufmann Publishers, Aug. 1995.

[27] Y. Guan, “Predictive modeling for risk analysis of wood packaging material,” tech. rep., Canadian Food Inspection Agency, Ottawa, ON, Canada, Aug. 2018.

[28] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[29] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.