Identifying Burns in Medical Device Reports

Lydia P. Gleaves and David A. Broniatowski

The George Washington University, Washington DC, 20052, USA
broniatowski@gwu.edu and lpgleaves@gwu.edu

Abstract. As part of ongoing post-market surveillance, the US Food and Drug Administration (FDA) accepts medical device reports (MDRs) describing possible device malfunctions. FDA analysts examine these reports individually, assessing whether a widespread device malfunction may be endangering public health. This work offers a classifier that identifies reports likely to involve a person being burned. The classifier uses a set of n-grams, taken from a large body of medical device report texts, as features; the problem's dimensionality is reduced using a principal component analysis. Three classifiers were then validated using a smaller body of reports annotated by a set of FDA analysts, and the best classifier was selected from the group. The results indicate that a simple support vector classifier with features based on raw n-gram frequencies is able to identify these serious burn cases with 96% recall.

Keywords: supervised machine learning, natural language processing, medical devices

1 Introduction

The FDA receives hundreds of thousands of medical device reports every year describing device malfunctions or failures [1]. These reports may be filed by patients, healthcare professionals, or device manufacturers, among others. To aid postmarket surveillance, FDA analysts are tasked with determining if these MDRs indicate a widespread device malfunction or failure endangering public health. Analyst responses often differ given the same MDRs, creating uncertainty in how to proceed. This uncertainty may have disastrous consequences, allowing dangerous devices to remain on the market or leading to life-saving devices being suspended for unrelated issues.

Machine learning classifiers have had many applications in medical decision-making [citation needed] and diagnosis [citation needed]. Here, they may be used to augment the MDR analysis process by flagging reports most likely to contain patient injuries. This work focuses on building a classifier capable of identifying whether a report conveys that a person has suffered a serious burn.

2 Description of Data

There are two important sets of data used in this work: the first is a large "complete" corpus of medical device reports filed with the FDA. There are over sixty-four thousand reports in this corpus, each containing the string "burn." The second is a smaller corpus of medical device reports, taken from the complete corpus and annotated in surveys in previous work. In those surveys, the text bodies of these 499 reports were individually assessed by five FDA analysts. Each analyst was asked to assess whether the report in question described a person suffering a serious burn injury, where "serious" was defined as life-threatening or permanent. They were given the options "Yes," "No," and "Not Enough Information" ("NEI"). These options are the classes that this paper will use, and the five votes will be used to determine a proxy "ground truth" when time to validate a model.

Of the annotated reports, the majority were categorized as "Yes" (49.7%) or "No" (45.9%), with few "NEI" assessments (4.4%). As there were more than two raters, inter-rater reliability was assessed using Fleiss' kappa [2]. This measure was found to be $\kappa = 0.65$, demonstrating "substantial agreement" [3].

Before beginning any analysis, a new "large" corpus was created by removing all reports in the smaller corpus from the complete corpus, still leaving approximately sixty-four thousand reports.

Each corpus was tokenized and cleaned before analysis, first removing stopwords and words of fewer than three characters, then lemmatizing and stemming each token.

3 Methods

3.1 Feature Extraction

Just as the analysts assessed the reports based on their text alone, so does each classifier. Because computers are as yet incapable of understanding and processing text in the same way that humans do, approximations must be made. This project uses the presence of sequences of words of length n, known as n-grams [4], [5], to characterize text. In Fig. 1. below, the sentence "The classifier was helpful" is split into unigrams (n = 1), bigrams (n = 2), and trigrams (n = 3).

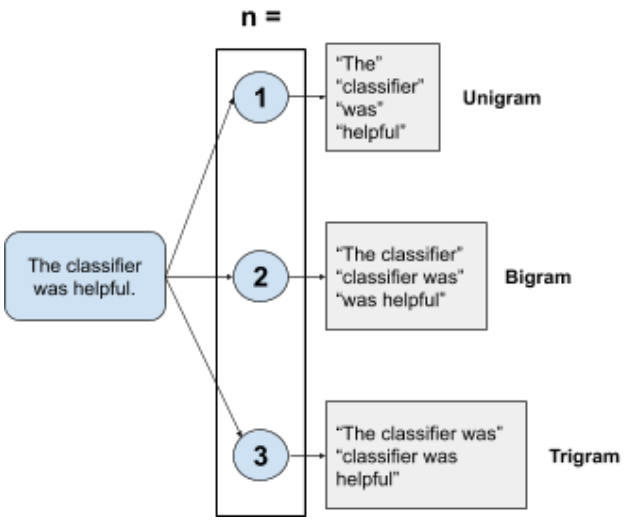

Fig. 1. Sentence split into unigrams, bigrams, and trigrams

When using n-grams as features, word sequences of length 2 or 3 have been shown to be the most effective [4]. To get the most out of this approach, this work uses word sequences of length 2 (bigrams) and 3 (trigrams), a decision modeled after others in the literature [5].

Classifier features were generated from the most common bigrams (occurred at least 500 times) and trigrams (occurred at least 300 times) in the large corpus. This produced 1,505 features of interest.

3.2 Feature Selection

With 1,505 features and 499 reports, a simple regression would be unwieldy and plagued by the curse of dimensionality; the features must be pared down to a subset that still adequately represent the data. The most popular way to reduce the dimensions of a dataset is likely principal component analysis [6]. A term-document matrix was created for the corpus, with each term being an n-gram feature and each document a device report, and a principal component analysis was performed on the normalized data. The resulting scree plot is shown in Fig. 2 below, zoomed to the first 25 components for clarity.

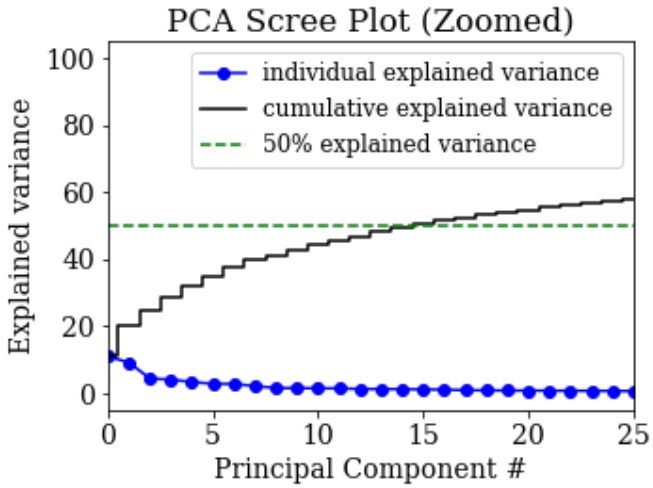

Fig. 2. Scree plot above, zoomed to between 0 and 25 principal components, offering a better view of the milestones

While 90% of the variance is explained in the first 157 (~10% of total) principal components, 80% of the variance is explained in the first 91 (~6%) components. In an effort to be as conservative as possible with computational power, the first 91 components will form the feature set of the classifiers built below. This follows the rule of thumb that one should have at least five samples per feature [7], with a ~5.5:1 ratio of features to samples here.

3.3 Classifier Fitting

As this is a supervised classifier, a "true" label, or ground truth, needed to be established for each report.

Each report was assessed five times; the ground truth label was approximated using the majority vote. As stated above, the analysts' inter-rater reliability was found to be $\kappa = 0.65$ using Fleiss' kappa. In the case of a tie, the more severe label was favored: "Yes" is the most severe, followed by "NEI," with "No" being the least severe and therefore never chosen in the case of a tie. Table 1 shows each tie case and its result.

Table 1. Tie cases.

Label 1	Label 2	Resulting Label
NEI	Yes	Yes
NEI	No	NEI
No	Yes	Yes

Each report receives its one label, following the tie severity table if necessary. With true labels for the data, the classifier construction can begin. This is a multinomial classification problem, as each document must be assigned one of three labels.

Three types of classifiers were built: a logistic regression, a Naïve Bayes classifier, and a support vector machine (SVM). Each classifier was tuned for "Yes" recall; as the goal of the work is to identify reports in which a person has been burned, it seemed most valuable to correctly identify as many of those as possible, forgiving to some extent lower precision.

Logistic Regression. The first classifier constructed is a multinomial logistic regression. This regression uses a one-vs-rest strategy, fitting a binary classifier to each label.

Naïve Bayes Classifier. The second classifier used is a multivariate Bernoulli Naïve Bayes classifier. Some Naïve Bayes classifiers represent a document by the set of word occurrences within that document, including the counts of those occurrences, while others represent a document by a vector of binary variables indicating whether a word occurs in the document, regardless of count [8]. This classifier is a twist on the latter, using the n-gram features rather than single words. The probability of a label is determined not only by the presence of certain words, but by the absence of others.

Support Vector Machine. The third classifier is a multiclass support vector machine, using C-support vector classification. This classifier uses the radial basis function kernel.

4 Results

4.1 Logistic Regression Results

The first classifier tested is the multinomial logistic regression; its classification report is shown in Table 2 below; it has an impressive "Yes" recall and the highest average accuracy, 0.7275, over 10-fold cross-validation.

Table 2. Classification Report, Logistic Regression

Label	Precision	Recall	F1-Score
NEI	1.00	0.55	0.71
No	0.92	0.88	0.90
Yes	0.88	0.96	0.92
Weighted Average	0.91	0.90	0.90

4.2 Naïve Bayes Classifier Results

The Bernoulli Naïve Bayes classifier came next; its classification report is in Table 3. With an average accuracy of 0.6299 in 10-fold cross-validation and lower precision and recall across the entire board, this classifier underperformed severely compared to the others. This may be due to poor hyperparameter selection for the task.

Table 3. Classification Report, Naïve Bayes

Label	Precision	Recall	F1-Score
NEI	0.19	0.36	0.25
No	0.75	0.70	0.72
Yes	0.75	0.73	0.74
Weighted Average	0.72	0.70	0.71

4.3 Support Vector Machine Results

The third classifier looked at is the RBF-kernel support vector machine, which had an average accuracy of 0.6789. It also matched the logistic regression in terms of "Yes" recall.

Table 4. Classification Report, Support Vector Machine

Label	Precision	Recall	F1-Score
NEI	1.00	0.50	0.67
No	0.93	0.88	0.91
Yes	0.87	0.96	0.91
Weighted Average	0.91	0.90	0.90

4.4 Overall Results

When comparing average cross-validation accuracies, in Fig. 3 below, again the Naive Bayes classifier can be set aside. Here, the logistic regression outshines the SVM, with a 15.5% relatively higher average accuracy. However, given that this accompanies a 27.9% increase in run time, the results favor the SVM. Choosing to trade accuracy for speed allows for easier scaling to larger datasets in the future.

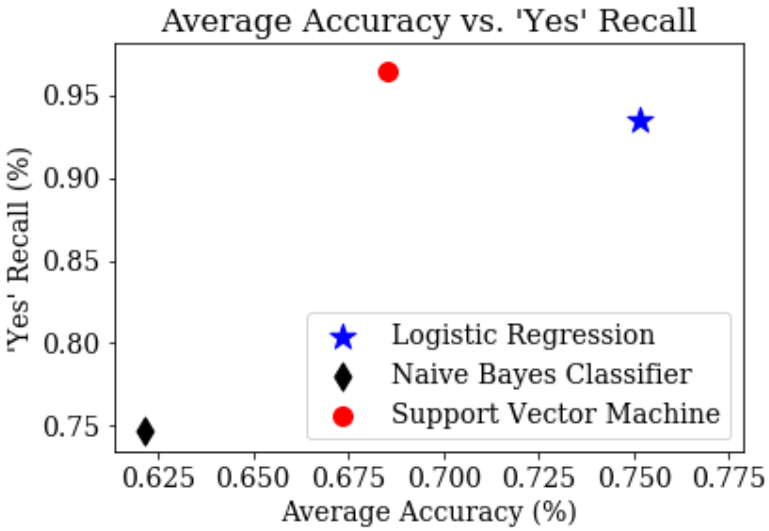

Fig. 3. Average accuracy vs. average "Yes" recall

5 Conclusions

5.1 Conclusions

With 96% recall for serious burns and a 68% accuracy overall, the SVM shown here is quite capable of identifying reports containing serious burns reliably. Most incorrect classifications tend towards more severe answers (e.g., classifying as "Yes" when the ground truth is "No"), which in practice is more appropriate than tending towards underrating the severity of a report. With this work, the foundation is laid for a tool that can aid analysts in identifying reports containing burn injuries. Future work should move past the pilot stage and scale the classifier to a larger set of burn reports.

It should be pointed out that this type of classifier may not generalize to all reports if the language surrounding other injuries is too dissimilar in structure. A "Yes" report in this dataset is likely to have the phrase "degree burn", for example; other injury reports may not have such clear signals. Future work should explore this method with different types of MDRs.

Additionally, the analyst assessments of the reports in the original survey dataset were based only on the text of the report. There may be other aspects of a report that influence their assessments in reality, making this a less than representative classifier. Future work should endeavor to identify these other aspects and meaningfully quantify them if possible.

5.2 Acknowledgement

The authors would like to thank Keith Walker for gathering the initial report texts and conducting the initial analyst survey.

References

1. FDA, Medical Device Reporting (MDR): How to Report Medical Device Problems, https://www.fda.gov/medical-devices/medical-device-safety/medical-device-reporting-mdr-how-report-medical-device-problems, last accessed 2019/05/10.
2. Fleiss, J. L.: Measuring nominal scale agreement among many raters. Psychology Bulletin 76(5), 378-382 (1971).
3. Landis, J. R., Koch, G. G.: The measurement of observer agreement for categorical data. Biometrics 33(1), 159-174 (1977).
4. Fürnkranz, J: A study using n-gram features for text categorization. Austrian Research Institute for Artificial Intelligence 3, 1-10 (1998).
5. Houvardas, J., Stamatatos, E.: N-Gram Feature Selection for Authorship Identification. In: Euzenat, J., Domingue, J. (eds.) Artificial Intelligence: Methodology, Systems, and Applications 2006, AIMSA, vol. 4183 Springer, Berlin, Heidelberg (2006).
6. Abdi, H., Williams, L. J.: Principal Component Analysis. In: Li, S. Z. (ed.) Encyclopedia of Biometrics. Springer, US (2012).
7. Pattern Recognition - 4th Edition, https://www.elsevier.com/books/pattern-recognition/theodoridis/978-1-59749-272-0, last accessed 2019/05/10.
8. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization, vol. 752, pp. 41-48 (1998).