Automated Coding of Worker Injury Narratives

Alexander C. Measure
U.S. Bureau of Labor Statistics
2 Massachusetts Avenue NE, Washington DC 20212 U.S.A.

Abstract
Much of the information about work related injuries and illnesses in the U.S. is recorded only as short text narratives on Occupational Safety and Health Administration (OSHA) logs and Worker's Compensation records. Analysis of these data has the potential to answer many important questions about workplace safety, but typically requires that the individual cases be "coded" first to indicate their specific characteristics. Unfortunately the process of assigning these codes is often manual, time consuming, and prone to human error.

This paper compares manual and automated approaches to assigning detailed occupation, nature of injury, part of body, event resulting injury, and source of injury codes to narratives collected through the Survey of Occupational Injuries and Illnesses, an annual survey of U.S. establishments that collects OSHA logs describing approximately 300,000 work related injuries and illnesses each year. We review previous efforts to automate similar coding tasks and demonstrate that machine learning coders based on the logistic regression and support vector machine algorithms outperform those based on naïve Bayes, and achieve coding accuracies comparable to or better than trained human coders.

Key Words: machine learning; statistical learning; natural language processing; text classification; logistic regression; naïve Bayes; support vector machines

1. Introduction

Much of the information about work related injuries and illnesses in the U.S. is recorded as short written narratives on OSHA logs and Worker's Compensation records. An injury to a nurse, for example, may ultimately be recorded as:

> Job title: *registered nurse*
>
> What was worker doing?
> *Employee was moving patient*
>
> What happened?
> *Patient became agitated and pushed employee causing her to fall on her wrist*
>
> What was the injury?
> *Sprained left wrist and contusions to left knee*
>
> What was the object or substance that inflicted the injury?
> *Floor and patient*

Analysis of this data is useful for safety surveillance and injury prevention, but is complicated by its unstructured nature. A common approach is to first assign predefined codes to each narrative to indicate characteristics of interest, and then to perform aggregate level analysis on the codes.

Unfortunately, the process of assigning these codes is typically manual, resource intensive, and vulnerable to human error. The Bureau of Labor Statistics, for example, which collects and codes OSHA logs describing approximately 300,000 incidents each year through its annual Survey of Occupational Injuries and Illnesses (SOII), requires an estimated 25,000 hours of labor for the initial coding task, and many additional hours to find and correct errors. Despite these considerable efforts, there is reason to believe that some errors go uncaught.

An important goal, therefore, is to improve both the quality and efficiency of coding. Progress in the field of natural language processing suggests that computers may be able to help by partially or fully automating the process. Previous research has identified two broad approaches to accomplishing this task, which we formulate more generally as text classification.

The first, the knowledge engineering approach, consists of manually encoding human knowledge into computer programs. To automatically assign occupation codes, for example, one might create a program made up of rules like the following:

If job title contains the word "janitor" then assign code 37-2011

Here, code 37-2011 corresponds to the Standard Occupational Classification (SOC) system's code for Janitors and Cleaners.

This works well for simple tasks; unfortunately many coding tasks are not simple. For example, the 10,325 Janitor and Cleaner cases collected for the 2011 Survey of Occupational Injuries and Illnesses included more than 2,000 distinct job titles and more than 80% of these cases had job titles that did not include the words "janitor" or "cleaner". This variability often requires many, and sometimes very complex rules to achieve high levels of automated coding performance.

An alternate approach, machine learning, avoids the problem of manually creating rules by using a computer to learn a model of code assignments directly from previously collected (and typically coded) data. One challenge is creating a representation of the relevant inputs that is amenable to modeling. A common approach is to represent each document (i.e. each narrative or case) with a vector where each element corresponds to a pre-determined feature of the data considered relevant for the classification task, and each value provides information about the state of that feature in a particular document. Text is typically incorporated into this vector using the *bag-of-words* approach, where each word is treated as a feature, and its occurrence in a particular document is indicated by its value. Once constructed, these vectors can be used with any of a wide variety of popular machine learning algorithms including logistic regression, naïve Bayes, decision trees, support vector machines, or neural networks.

A natural question is which approach works better, knowledge engineering or machine learning? The truth is that in practice both approaches are frequently combined to varying degrees. Still, there is a legitimate question as to which provides better performance for

the requisite costs. Creecy et al. explored these differences in work comparing a knowledge engineering and a machine learning approach to assigning industry and occupation codes to Census narratives. They found the knowledge-based approach not only required vastly more labor to implement (192 person-months compared to 4 person-months for machine learning), but also had worse performance [1]. This does not mean knowledge-based approaches are always worse. In fact, they have occasionally performed very well on tasks where elaborate knowledge based resources have already been constructed and training data is limited, such as a medical text classification challenge (CMC 2007) focused on coding radiology reports [2]. In general, however, the knowledge engineering approach becomes increasingly unattractive as the complexity of the classification task and the availability of training data and computing power increase. As a result, knowledge-based approaches have increasingly lost popularity since the early 1990's and receive little attention from modern text classification researchers [3].

Recent efforts to automatically classify worker injury narratives have focused almost exclusively on machine learning. Lehto and Wellman used the naïve Bayes algorithm to automatically assign 1 of 19 "event" codes to workers' compensation narratives [4]. Similarly, Bertke et al. used naïve Bayes to automatically classify workers compensation narratives into 1 of 3 categories [5]. Both report relatively good results, but their studies leave a number of important problems unresolved.

Perhaps the biggest is that the quality of their automated coding still lags behind that of their human coders. For organizations where coding quality is of high importance and manual methods are available, this is a serious barrier to adoption.

Another problem is that the level of coding detail required for many surveillance tasks is much higher than that pursued in Lehto and Bertke's work. For the Survey of Occupational Injuries and Illnesses, for example, the Bureau of Labor Statistics assigns detailed occupation, nature of injury, part of body, event resulting in injury, and source of injury codes to each case, and each of these categories has hundreds of potential classifications. Source alone has 1,400 codes, and occupation, 800. For organizations like the Bureau of Labor Statistics to adopt these methods, they must demonstrate high effectiveness on the very broad and complex coding tasks they face.

The goal of this study therefore is twofold: to improve the quality of automated coding by exploring alternative machine learning approaches, and to test the feasibility of automatically assigning codes at the high level of detail and breadth of scope required for the Survey of Occupational Injuries and Illnesses.

2. Methodology

The data for our experiments comes from the nearly 300,000 cases collected and coded for the 2011 Survey of Occupational Injuries and Illnesses. We removed cases from Puerto Rico, which tend to be in Spanish, all cases where no job title was reported, all cases without a response to at least one of the injury narrative questions, and all cases that were considered unusable for the purposes of SOII estimation. Of the remaining cases, we randomly selected 261,000 and then randomly divided these into 3 data sets; a training set of 195,000 cases, a validation set of 65,000 cases, and a test set of 1,000 cases.

As part of the Bureau's normal collection activities, each case had been assigned 5 codes by Bureau of Labor Statistics employees. These include an occupation code, assigned according to the 2010 version of the Standard Occupational Classification system, and a nature, part, event, and source code assigned according to version 2.01 of the Occupational Injuries and Illnesses Classification system. Each of these codes went through the Bureau's normal review process which includes automated checks for invalid code combinations and manual reviews in BLS regional and national offices.

In addition to these codes, each case had the following potentially classification-relevant information associated with it:

- a short narrative indicating the worker's occupation title
- a checkbox indicating the worker's occupation category (1 of 12 possible, including "other")
- a short narrative indicating the occupation category, if the respondent indicated "other"
- a narrative answering "What was the employee doing before the incident occurred?"
- a narrative answering "What happened?"
- a narrative answering "What was the injury or illness?"
- a narrative answering "What object or substance directly harmed the employee?"
- the ownership of the worker's establishment (private, state government, or local government)
- the 2007 North American Industry Classification System (NAICS) code for the establishment
- the state where the establishment was located

To measure the performance of human coders we hid the codes assigned to the 1,000 cases in the test set and then distributed these cases among BLS trained coders in such a way that each case was assigned to 3 different people. Coders were then instructed to assign codes as accurately as possible without referencing previously coded data or discussing code assignments with others. Accuracy was calculated in the usual way, as the number of code assignments matching the original code divided by the total number of possible code assignments.

Three machine learning algorithms were chosen for comparison based on their popularity for the text classification task; naïve Bayes, regularized logistic regression, and support vector machines. We used the free and open-source implementations of these algorithms available through the scikit-learn 14.1 software package [6]. We describe each briefly below.

Naïve Bayes

Let y denote a specific classification (for example, that the injury event is a "fall"), let n denote the number of features in our feature representation, and let $x_1, \ldots, x_n$ denote the values of all features associated with a given document. Naïve Bayes performs classification by first estimating the probability of classification y given observed feature values x_1 through x_n using Bayes' theorem, which can be written as:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n|y)}{P(x_1, \ldots, x_n)} \quad (1)$$

$P(y)$ is typically estimated as the relative frequency of classification y in the training data, and $P(x_1, \ldots, x_n)$ can be ignored altogether since it is the same for all possible classifications, but $P(x_1, \ldots, x_n|y)$ is more problematic.

In practice there is never enough data to calculate a reasonable estimate of $P(x_1, \ldots, x_n|y)$. Instead, we make the "naïve" assumption that for a given classification y, the probability of observing any feature x_i is independent of observing any other feature. This allows us to calculate $P(x_1, \ldots, x_n|y)$ as $\prod_{i=1}^{n} P(x_i|y)$.

$P(x_i|y)$ can be calculated in multiple ways depending on our assumptions of the underlying distribution. In our experiments we try two popular variations, multinomial naïve Bayes, and Bernoulli naïve Bayes.

In multinomial naïve Bayes, $P(x_i|y)$ is estimated as

$$P(x_i|y) = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (2)$$

where N_{yi} denotes the sum of x_i over all training examples where the classification is y, N_y denotes the sum of all features $x_1, \ldots, x_n$ over all training examples with a classification of y, and α is a positive number that acts to smooth the maximum likelihood estimate of $P(x_i|y)$ to prevent zero probabilities. Smoothing is desirable because a single zero probability in the $P(y)\prod_{i=1}^{n} P(x_i|y)$ expression effectively erases whatever information was conveyed by the other terms.

In Bernoulli naïve Bayes, the value of x_i is restricted to either 0, or 1, and indicates the presence or absence of that feature. Here, $P(x_i|y)$ is calculated as

$$P(x_i|y) = P(i|y)x_i + (1 - P(i|y))(1 - x_i) \quad (3)$$

where

$$P(i|y) = \frac{M_{yi} + \alpha}{M_y + \alpha n} \quad (4)$$

and M_{yi} is the number of training examples with classification y containing feature x_i, and M_y is the number of training examples with classification y.

Classification is performed by assigning the value of y that maximizes the expression: $P(y)\prod_{i=1}^{n} P(x_i|y)$ [6–8].

Logistic Regression

Let S denote the set of possible classifications, let x denote a feature vector of length n + 1, where the first position is always 1, and let w_y denote a weight vector of length n + 1 specific to classification y. In the multi-class setting, logistic regression models the probability of classification y given feature vector x as

$$P(y|x) = \frac{e^{w_y^T x}}{\sum_{k \in S} e^{w_k^T x}} \quad (5)$$

The optimal weight vectors are calculated using convex optimization techniques that maximize the likelihood of the training data, subject to a penalty on the size of the weight vectors (also known as regularization) to prevent overfitting.

If we denote the number of training examples in our training set as m, and define y_i to have a value of 1 if the label of the i-th training example is classification y, and -1 otherwise, we can write this mathematically as

$$argmin_w \frac{1}{2} w^T w + C \sum_{i=1}^{m} \log(1 + e^{-y_i w^T x_i}) \quad (6)$$

Here, $\sum_{i=1}^{m} \log(1 + e^{-y_i w^T x_i})$ is the empirical loss, i.e. the penalty for not classifying the training data correctly, $\frac{1}{2} w^T w$ is the L2 regularization loss used to prevent overfitting, and C is a regularization constant which controls the tradeoff between minimizing the empirical loss and minimizing the regularization loss [9,10].

Once the weights have been learned, classification is performed by selecting the classification which maximizes $P(y|x)$.

Support Vector Machines

To perform classification with support vector machines we begin by first finding the hyperplane defined by $w^T x = 0$ for each possible classification y that maximizes the margin between the closest training example belonging to classification y, and the closest training example belonging to any other classification.

The weight vectors parameterizing these hyperplanes are calculated using convex optimization techniques that simultaneously minimize the squared hinge loss and the regularization loss (L2 loss in our experiments). We write this mathematically as

$$argmin_w \frac{1}{2} w^T w + C \sum_{i=1}^{m} \max(1 - y_i w^T x_i, 0)^2 \quad (7)$$

Once the weights have been calculated, classification is performed by calculating $w_y^T x$ for each possible classification y, and choosing the classification which produces the largest value [10].

Feature Representation

For each classification task, two feature representations were created: a baseline representation designed to include all obviously relevant information in a simple manner, and a "best" representation developed by manually and iteratively training models on different representations of the training set until settling on one that produced the most accurate model, as measured against the validation data set. Because the logistic regression and support vector machine algorithms demonstrated substantially better performance with the simple representation, and because they are widely considered

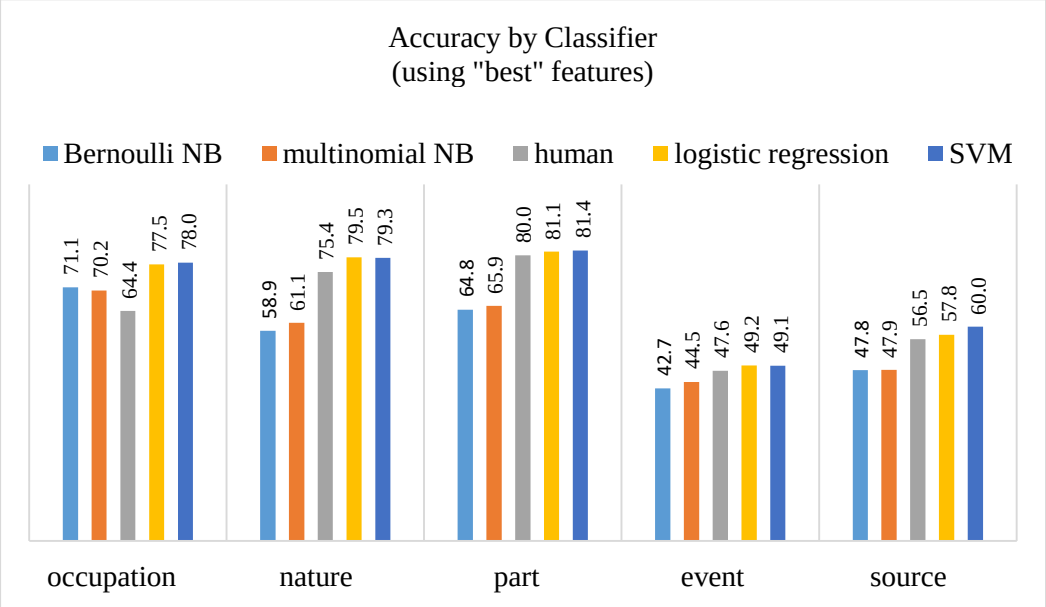
better classifiers than naïve Bayes for this type of task [3,11,12], the "best" representation was designed primarily to maximize their performance. The details of each representation are summarized in table 1.

Table 1. Feature Representations

Classification Task	Baseline Representation	Best Representation
occupation	job_unigrams, other_category_unigrams, category, naics	job_unigrams, job_unigrams + job_category, job_unigrams + naics2, job_unigrams + ownership, job_bigrams, other_category_unigrams, fips_state_code, naics
nature	incident_unigrams	nature_unigrams, incident_bigrams, naics
part	incident_unigrams	nature_unigrams, incident_bigrams, incident_trigrams
event	incident_unigrams	incident_unigrams, incident_bigrams, naics
source	incident_unigrams	source_unigrams, incident_bigrams, naics2, fips_state_code
See the Appendix for a more detailed description of each feature		

All three learning algorithms have hyper-parameters (parameters not directly learned from the training data) that must be set in some way; the regularization parameter for logistic regression and support vector machines, and the smoothing parameter for naïve Bayes. We selected separate hyper-parameter values for each combination of classification task and feature representation by iteratively training each algorithm using a range of hyper-parameter values and then choosing the values that produced the most accurate model, as evaluated against the validation data. The final models were then trained on the combination of the training set and validation set, and evaluated based on their accuracy on the test set. Table 2 contains the resulting accuracy scores for the machine learning algorithms and the human coders.

Table 2. Classification Accuracy (%)						
features	classifier	occupation	nature	part	event	source
	human	64.4	75.4	80.0	47.6	56.5
baseline	Bernoulli NB	68.7	62.7	62.4	42.0	43.8
baseline	multinomial NB	68.9	62.5	59.9	42.4	43.1
baseline	logistic regression	73.2	78.3	75.8	47.8	55.1
baseline	SVM	73.9	78.4	76.4	47.0	56.5
best	Bernoulli NB	71.1	58.9	64.8	42.7	47.8
best	multinomial NB	70.2	61.1	65.9	44.5	47.9
best	logistic regression	77.5	**79.5**	81.1	**49.2**	57.8
best	SVM	**78.0**	79.3	**81.4**	49.1	**60.0**
SVM = support vector machines, NB = naïve Bayes The highest accuracy for each classification task is shown in bold						



3. Discussion

Our results suggest two important findings. First, machine learning techniques like support vector machines and logistic regression can produce more accurate coders than naïve Bayes. Second, these more effective machine learning approaches can achieve classification accuracies similar to, or better than those achieved by human coders, even on highly detailed coding tasks like those performed for SOII.

Acknowledgements

The opinions presented in this paper are those of the author and do not represent the opinions or policies of the Bureau of Labor Statistics or any other agency of the U.S. Department of Labor.

References

1 Creecy RH, Masand BM, Smith SJ, *et al.* Trading MIPS and Memory for Knowledge Engineering. *Commun ACM* 1992;**35**:48–64. doi:10.1145/135226.135228

2 Pestian JP, Brew C, Matykiewicz P, *et al.* A shared task involving multi-label classification of clinical free text. In: *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. Association for Computational Linguistics 2007. 97–104.http://acl.ldc.upenn.edu/W/W07/W07-10.pdf#page=113 (accessed 27 Mar2014).

3 Sebastiani F. Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 2002;**34**:1–47.

4 Lehto M, Marucci-Wellman H, Corns H. Bayesian methods: a useful tool for classifying injury narratives into cause groups. *Inj Prev* 2009;**15**:259–65. doi:10.1136/ip.2008.021337

5 Bertke SJ, Meyers AR, Wurzelbacher SJ, *et al.* Development and evaluation of a Naïve Bayesian model for coding causation of workers' compensation claims. *J Safety Res* 2012;**43**:327–32. doi:10.1016/j.jsr.2012.10.012

6 Pedregosa F, Varoquaux G, Gramfort A, *et al.* Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research* 2011;**12**:2825–30.

7 Manning CD, Raghavan P, Schütze H. *Introduction to information retrieval*. Cambridge university press Cambridge 2008. http://www-nlp.stanford.edu/IR-book/ (accessed 27 Mar2014).

8 Metsis V, Androutsopoulos I, Paliouras G. Spam filtering with naive bayes-which naive bayes? In: *CEAS*. 2006. 27–8.http://classes.soe.ucsc.edu/cmps242/Fall09/lect/12/CEAS2006_corrected-naiveBayesSpam.pdf (accessed 27 Mar2014).

9 Ming-Wei C. Introduction to Logistic Regression and Support Vector Machine. http://l2r.cs.uiuc.edu/~danr/Teaching/CS446-12/Lectures/LR-SVM-slides.pdf (accessed 26 Mar2014).

10 Fan R-E, Chang K-W, Hsieh C-J, *et al.* LIBLINEAR: A Library for Large Linear Classification. *J Mach Learn Res* 2008;**9**:1871–4.

11 Ng AY, Jordan MI. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems* 2002;**2**:841–8.

12 Manning C, Klein D. Maxent Models, Conditional Estimation, and Optimization Without Magic. 2003.http://www.cs.berkeley.edu/~klein/papers/maxent-tutorial-slides.pdf (accessed 27 Mar2014).

Appendix

Feature	Description
fips_state_code	Code indicating the state in which the worker's establishment was located
naics	The 2007 North American Industry Classification System (NAICS) 6 digit code for the worker's establishment
naics2	The first 2 digits of the NAICS code of the worker's establishment
job_unigrams	Individual words from the job title narrative
job_unigrams + estab_ownership	Concatenation of individual words from the job title narrative and the code indicating the ownership of the worker's establishment
job_unigrams + job_category	Concatenation of individual words from the job title narrative and the code indicating the job category
job_unigrams + naics2	Concatenation of individual words from the job title narrative and the first 2 digits of the establishment's NAICS code
job_bigrams	Two word sequences from the job title narrative
other_category_unigrams	Individual words from the "other job category" text field
incident_unigrams	Individual words from the 4 narratives describing the circumstances of the incident
incident_bigrams	Two word sequences from the 4 narratives describing the circumstances of the incident
incident_trigrams	Three word sequences from the 4 narratives describing the circumstances of the incident
nature_unigrams	Individual words occurring only in the narrative corresponding to "What was the injury or illness?"
source_unigrams	Individual words occurring only in the narrative corresponding to "What object or substance directly harmed the employee?"