Data Science and Engineering (2018) 3:221–231
https://doi.org/10.1007/s41019-018-0071-7

Automatically Detecting Errors in Employer Industry Classification Using Job Postings

Alan Chern[1] · Qiaoling Liu[1] · Josh Chao[1] · Mahak Goindani[2] · Faizan Javed[1]

Received: 16 January 2018 / Revised: 8 June 2018 / Accepted: 22 July 2018 / Published online: 19 August 2018

Abstract
In the recruitment domain, knowing the employer industry of jobs is important to get an insight about the demand in each industry. The existing system at CareerBuilder uses an employer name normalization system and an employer knowledge base (KB) to infer the employer industry of a job. However, errors may occur during the computation of the job employer and in the construction of the employer KB with the industry attributes. Since the KB is huge, it is not possible to manually detect the errors. Therefore, in this paper we use machine learning techniques to automatically detect the errors. With the observation that the main jobs posted by an employer often relate to the employer industry, e.g., truck driver jobs often correspond to employers in the transportation industry, we develop a system that classifies the industry of an employer using job posting data. We aggregate job postings from an employer and derive features from employer names, employer descriptions, job titles, and job descriptions to predict the industry of the employer. Two models are used for classification: (1) support vector machine and (2) random forest. Our experiments show that random forest is more effective than SVM in identifying the errors in the existing industry classification system, which achieves precision 0.69, recall 0.78, and f-score 0.73. It especially better handles mixed feature vectors when normalization errors occur. We also observe that generally our models perform better in detecting errors for industries that have higher error rates.

Keywords Employer industry classification · Job postings · Multiclass classification · Error detection

1 Introduction

Different employers belong to different industries such as Transportation and Warehousing (Transportation), and Health Care and Social Assistance (Health Care), etc., according to the North American Industry Classification System (NAICS)[1]. Classifying the industry of employers has several applications in the recruitment domain. Knowing the industry of an employer helps to get an insight about the demand in each industry, such as the number of jobs and top job posters. This can be useful for labor market analysis since we can know which industries are important and provide more jobs.

The existing system at CareerBuilder[2] uses an employer name normalization system [14–16] and an employer knowledge base (KB) to infer the employer industry of a job. The employer name normalization system normalizes employer names and links them to an employer entity in the KB. The employer KB contains an industry attribute for each employer entity. However, errors may occur during the computation of the job employer and in the construction of the employer KB with the industry attributes. For example, the industry of the employer "First Transit"

✉ Qiaoling Liu
qiaoling.liu@careerbuilder.com

Alan Chern
alan.chern@careerbuilder.com

Josh Chao
josh.chao@careerbuilder.com

Mahak Goindani
mgoindan@purdue.edu

Faizan Javed
faizan.javed@careerbuilder.com

[1] CareerBuilder, Norcross, GA, USA

[2] Purdue University, West Lafayette, IN, USA

[1] https://www.census.gov/eos/www/naics/.

[2] http://www.careerbuilder.com/.

Table 1 Distribution of 9723 companies in each industry (according to 2017 NAICS)

Sector	Description	Percentage
54 or 56	Professional, scientific, and technical services; Administrative and support and waste management and remediation services	27.59
62	Health care and social assistance	13.70
31–33	Manufacturing	9.33
61	Educational services	7.05
44–45	Retail trade	6.61
48–49	Transportation and warehousing	6.43
52	Finance and insurance	5.75
72	Accommodation and food services	3.94
Other	The rest sectors, including 51, 42, 81, 53, 92, 23, 71, 55, 22, 21, 11	19.60

should be Transportation, but is incorrectly labeled as "Office Administrative Services" in the KB. The KB is huge with about 20M entities, and hence, it is not possible to manually detect the errors.

We observe that some of these errors can be corrected with the help of job posting data since the main jobs posted by an employer often relate to its industry. For example, truck driver jobs often correspond to employers that belong to transportation industry. When we look at the jobs posted by the employer named "First Transit," the titles of the jobs posted include "Shuttle Driver," "Bus Operator," "Automotive Technician". By looking at these job titles, we can see that they belong to the Transportation domain. Also, the word "Transit" in the employer name is a good indicator that the employer industry is Transportation. Employers named "Fresenius Medical Care," "Community Health Systems, Inc." contain keywords "medical", "health" and belong to the Health Care industry.

Therefore, in this paper we propose to use the job postings to classify the industry of an employer, with the goal of automatically detecting the errors in the existing system. We aggregate job postings from an employer and derive features from employer names, employer descriptions, job titles, and job descriptions from the job postings for classifying the employer industry. We use machine learning models such as support vector machine (SVM) and random forest to learn the signals. Through experiments, we observe that random forest is more effective than SVM in identifying the errors in the existing industry classification system, which achieves precision 0.69, recall 0.78, and f-score 0.73. It especially better captures the interactions between different signals in more complex and mixed feature vectors, when normalization errors occur. We also observe that generally for industries that have higher error rates, our models perform better in detecting errors, according to a moderate correlation between the f-scores and the error rates for different industries.

The outline of the rest of the paper is as follows. We describe how we collect our dataset in Sect. 2 and present our ideas and the methods used in Sect. 3. We describe the evaluation metric, experiments, and results in Sect. 4, followed by discussion and scope for future work in Sect. 5. Related work is discussed in Sect. 6. Finally, we close with concluding remarks in Sect. 7.

2 Dataset

We used the Jobfeed API[3] to get the job posting data. First, we collected the top 10,000 normalized employers that provide the most number of jobs during a one-year period from the beginning of November 2016 to the end of October 2017. We used one year as the time frame to prevent the job count distribution to be biased due to seasonal fluctuations. For each of the normalized employers, the API also returns its industry sector, based on the definition from 2017 NAICS [1]. Therefore, these industry sectors are used as our categories for classification. The percentage of employers for each industry is shown in Table 1. Note that we filter out employers that do not have a valid industry label based on the API, which leaves 9723 employers. We also combine sector 54 (Professional, Scientific, and Technical Services) and sector 56 (Administrative and Support and Waste Management and Remediation Services) because the nature of companies in these two industries tend to overlap. A prominent example is recruitment consultancy agencies that can be equally categorized as sector 54 as well as sector 56. Finally, we create an "Other" category to combine all the other sectors that have fewer employers than sector 72 (Accommodation and Food Services) does.

[3] https://us.jobfeed.com/api/v3/help.

After this, we divided the data into three parts: 80% for training, 10% for validation, and 10% for testing, using random sampling. We observed that the percentage of employers in the training set, validation set, and test set is similar as that in the complete dataset. We can see from Table 1 that there is a lot of imbalance in the proportion of examples between the eight industries, with the highest being 27.59% examples in the combined sector 54 or 56 and the lowest being 3.94% examples in sector 72.

Since we collect the industry class labels from the API, which are computed using the existing employer name normalization system and the employer KB at CareerBuilder, we call them *legacy labels*. We are aware that such legacy labels may contain errors which occur during the computation of the job employer and the construction of the employer KB with the industry attributes. Yet, we believe that the majority of the legacy labels would be correct, and using them for learning signals from job postings would be sufficient. This is supported by that our randomly sampled test set has an error rate of 31.9%, as shown in Table 3.

3 Methods

Based on the above dataset, we build multiclass classifiers for the nine industry sectors. In the following, we will first describe how we generate the features and then introduce the machine learning models we used.

3.1 Feature Generation

For each employer, we generate five types of features: the normalized titles of the jobs posted by this employer, the keywords in the employer description, the keywords in the employer name, the entropy calculated from the normalized titles, and the matching rate of staffing related keywords and phrases within the job descriptions.

Using the Jobfeed API, we get a list of employers, along with the employer description for each, and the normalized job titles [12] and job descriptions corresponding to the jobs posted by them.

The job titles were normalized via a job title normalization system [12], which built a hierarchical job title taxonomy by first classifying job postings into Standard Occupational Classification (SOC)[4] majors and then clustering a large number of job postings within each SOC major. During normalization, it first predicted the SOC major for a query job title via SVM and then used kNN to compute the final cluster and normalized title. More details about the job title normalization system can be found in [12].

Each employer posts various jobs: technical as well as non-technical, like software engineer, truck driver, sales manager, customer representative. We assume that the most frequent jobs posted by an employer are technical jobs, i.e., they are related to the industry of the employer. And, using the infrequent jobs that are uninformative might not be useful for classification and can add noise. So we filter out the infrequent jobs that are uninformative in the following way.

Let T be the list of all 5426 normalized titles. For each normalized title $t \in T$ and an employer e, we first calculate a ratio score $R_{e,t}$ based on the percentage of jobs posted by e that have title t. Next, we define a list T_e for each employer e that consists of those normalized titles that have their ratio score greater than or equal to 0.01, that is,

$$T_e = \{t | R_{e,t} \geq 0.01\} \quad (1)$$

The threshold of 0.01 was chosen empirically so that T_e contains the main normalized titles for employer e that are related to the employer industry while excluding some noisy infrequent jobs that are mostly non-technical and unrelated to the employer industry. As shown in Fig. 1, the number of nonzero features decreases with increasing threshold, and the difference is clearly the most significant between a threshold of 0 and 0.01, whereas the change starts to flatten out from then on. Thus, we chose to filter out the noisy features that have a very low ratio score below 0.01.

After this we calculate the vocabulary of normalized titles for each industry. There are certain normalized titles that appear in various industries such as sales manager, customer representative, and these have high frequency in multiple industry sectors. We want to exclude such titles

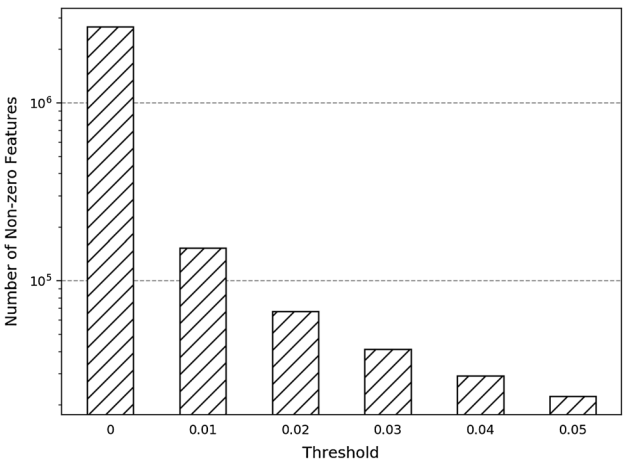

Fig. 1 Number of nonzero features generated with various job title ratio score thresholds

[4] https://www.bls.gov/soc/.

Table 2 Top job titles, employer description keywords, and company name keywords for three industries

Feature type	Vocabulary size	Examples
(1) Health care and social assistance industry (sector 62)		
Job titles	261	Dialysis technician, assisted living supervisor, inpatient services nurse, rehabilitation aide, triage nurse
Employer description keywords	82	Hospitalized, outpatients, dietitian, symptom, prostate
Company name keywords	131	Hospice, clinic, rehabilitation, hospital, medical, health
(2) Retail trade industry (sector 44–45)		
Job titles	311	Cashier (office and administrative support), customer service assistant, shipping and receiving clerk, loss prevention manager (protective service), warehouse unloader
Employer description keywords	223	Retailer, prices, supermarket, merchandise, clothing
Company name keywords	46	Retail, food, market, grocery, shop, store, mart
(3) Transportation and warehousing industry (sector 48–49)		
Job titles	123	Truck driver, commercial driver's license (CDL) driver, independent contractor ((transportation and material moving)), flatbed driver
Employer description keywords	66	Transporting, fleets, driver, hauling, trucks, passengers
Company name keywords	40	Carriers, airlines, trucking, transport, freight, transfer

since they do not provide any discriminative information and can add noise to the system. Selecting the important titles for the vocabulary manually is a very tedious and time-consuming task, given that we have 5426 possible normalized titles. Hence, we came up with a significance score which helps to select the significant titles without any manual effort. The idea is to provide more importance to the titles that only belong to specific individual sectors than to the ones spanning multiple sectors. For each industry i, and for each normalized title t, we calculate the significance score as:

$$S_{ti} = \frac{f_{ti}}{f_t} \tag{2}$$

where f_{ti} is the frequency of the normalized title t across all employers in the industry i, and f_t is the frequency of the normalized title t across all employers. After this, to form the vocabulary V_{ti} of normalized titles for an industry i, we select those normalized titles that have their significance score and frequency greater than certain thresholds.

$$V_{ti} = \{t | S_{ti} \geq \theta_{si}, f_t \geq \theta_{fi}\} \tag{3}$$

Here, θ_{si} is the threshold on the significance score for industry i, and θ_{fi} is the threshold on the frequency of title for industry i. The values of θ_{si} and θ_{fi} should be chosen such that V_{ti} has a reasonable size and is relevant to the industry i. Hence, these values can be different for different industries. It is a very tedious process to manually decide appropriate values of thresholds for each industry. Hence, we experimented with using the quartiles as the thresholds. We first filter out normalized titles that have $f_{ti} \leq 1$. For the remaining normalized titles, we compute the median of S_{ti} and the median of f_t, and use them as θ_{si} and θ_{fi}, respectively. Then, we check the vocabulary size obtained after applying these thresholds, and if the size is less than 50 (too few titles), we reduce θ_{fi} to the first quartile of f_t, while keeping θ_{si} unchanged. This is done in order to increase the number of relevant titles in the vocabulary. Table 2 shows the size of V_{ti} and some example titles in V_{ti} for a few industries.

Now, we can create a title feature vector for each employer. For each $t \in V_{ti}$ and an employer e, the feature value $v_{e,t}$ is given as:

$$v_{e,t} = \begin{cases} R_{e,t} & t \in T_e \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We use $R_{e,t}$ as the feature value in order to assign more importance to the more frequent jobs posted by an employer.

This process is repeated with a slight modification using the employer descriptions we collected to generate a vocabulary V_{di} of employer description keywords for each industry i. Here, because the set of distinct words that makes up the employer descriptions contains significantly more words compared to the number of normalized job

titles, instead of using the median to choose the threshold value, we use the 90th percentile in order to reduce the size of V_{di}. Table 2 shows the size of V_{di} and some example keywords in V_{di} for a few industries.

Next, for each word $d \in V_{di}$ and an employer e, we calculate a ratio score $R_{e,d}$ based on the frequency of d in the employer description we collected for employer e, normalized by the total number of words in the employer description of e. The employer description feature vector $v_{e,d}$ is then formed based on V_{di} for each employer as:

$$v_{e,d} = \begin{cases} R_{e,d} & d \in D_e \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where D_e is the list of words present in the employer description for employer e.

In a similar way, for each industry i, we generate a vocabulary V_{ni} of keywords present in employer names. There are two types of employer names: Unnormalized Names and Normalized Names. Unnormalized Names are the raw names (mentions) of an employer extracted from the job postings. These names are then given as input to the normalization system and are mapped to entities in the employer KB. The name of this normalized entity is called the Normalized Name. By employer name, we refer to the set of all Unnormalized Names and the Normalized Name of an employer. For each industry, we define a significance score for the unigrams present in the employer names. Then to form the vocabulary of keywords for that industry, we select those unigrams that have their significance score and frequency greater than certain thresholds. The thresholds for V_{ni} are chosen in a similar way as for how we compute V_{ti} and V_{di}. Table 2 shows the size of V_{ni} and some example keywords in V_{ni} for a few industries.

Now, we can create a keyword feature vector for each employer. For each $w \in V_{ni}$ and an employer e, the feature value $v_{e,n}$ is given as:

$$v_{e,n} = \begin{cases} 1 & n \in N_e \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where N_e is the list of unigrams present in the employer name for employer e.

With the normalized titles, we also calculate an entropy value for each employer to capture the variation of the various types of job posts. We believe that this feature would be helpful for distinguishing staffing companies (i.e., sector 56) because these companies tend to hire for a more diverse range of jobs. The job title entropy for an employer e is calculated as:

$$H_e = -\sum_{t \in T} R_{e,t} \ln R_{e,t} \quad (7)$$

In order to further identify the staffing companies, we look for certain keywords and key phrases within the job descriptions associated with the job postings of each employer. For example, we believe job descriptions that contain words such as "client" and phrases such as "is looking for" are more likely to indicate a job posted by a staffing company. Specifically, we first create a list of such keywords and key phrases. Then for each employer, we extract 100 job descriptions and calculate the percentage of those that match at least one keyword or key phrase from the list. We call this feature the job description staffing rate of the employer.

The combined feature vector for employer e will be the concatenation of the job title feature vector $v_{e,t}$, the employer description keyword feature vector $v_{e,d}$, the employer name keyword feature vector $v_{e,n}$, the job title entropy value H_e, and the job description staffing rate.

3.2 Learning Algorithms

We use two well-known models for our classification task: (1) support vector machine (SVM) [11, 21] and (2) random forest [4, 8, 11].

Specifically, we used the LIBSVM tool [6] to build the SVM Model, and we used the Spark ML package[5] to build the random forest model. As data sampling has been shown in [3, 7] to improve the random forest classification performance, for the training of our random forest model, we oversampled each class by duplicating the examples to match the class with the greatest number of examples. This reduces the imbalance due to the distribution of the employers within different industries.

4 Experiments

4.1 Disagreement Set

Our key idea of using machine learning methods to automatically detect errors in the legacy labels is based on an assumption: when a model prediction disagrees with a legacy label, it has a higher chance to be an error. We call the set of such examples from test data the *Disagreement Set*. In other words, a legacy label is less likely to be wrong when all the model predictions agree with it. We call the set of such examples from test data the *Agreement Set*.

To check how reasonable this assumption is, we manually labeled the industry sectors for the whole test set. We observed that the error rate of legacy labels in the agreement set is 4.7% while in the disagreement set it is 59.4%, as shown in Table 3. This shows a much higher error rate

[5] https://spark.apache.org/docs/1.6.1/ml-classification-regression.html.

Table 3 Error rate in disagreement set and agreement set

	Size	Errors	Error rate (%)
Test set	973	310	31.9
Disagreement set	483	287	59.4
Agreement set	490	23	4.7

Table 4 Distribution of errors in test set

Error type	Count (%)
Incorrect name extraction	74 (23.9%)
Incorrect name normalization	110 (35.5%)
Incorrect KB attribute	126 (40.7%)

of legacy labels in the disagreement set. Therefore, we can say that this assumption is reasonable.

4.2 Error Types

We analyzed all the errors in the test set and observed three types of errors in the results of the legacy system.

1. *Errors due to incorrect name extraction* These are the errors in the extraction process (parsing) of the Unnormalized Name of the employer. For example, "Truck Driver Cdl-a" is an Unnormalized Name given as input to the normalization system. But there is no valid employer named "Truck Driver Cdl-a".
2. *Errors due to incorrect name normalization* These are the errors when the Unnormalized Name of an employer is valid (correct name extraction), but is incorrectly mapped to a wrong entity in the KB during the normalization process. For example, the Unnormalized Name "Omni Specialized LLC." is incorrectly mapped to a wrong entity "OWI Specialized, Inc." by the normalization system.
3. *Errors due to incorrect KB attribute* These are the errors when the extracted name is valid and its normalization is correct, but the industry attribute of the employer in the KB is marked incorrectly. For example, the Unnormalized name "Hornady Transportation" is a valid employer and is normalized to "Hornady Transportation LLC," which is a transportation company. But its industry in the KB is incorrectly labeled as "Temporary Help Services," which belongs to sector 56.

The distribution of each type of errors is presented in Table 4. We can see that while the smallest proportion of the errors is due to incorrectly extracted Unnormalized Names, the largest proportion of the errors is caused by incorrectly labeled industry attributes in the KB.

4.3 Metrics

To measure how useful the machine learning models are in detecting errors in the legacy labels, we use a set of utility metrics. Let E_T be the set of true errors that are manually verified in the test set. Let E_M be the disagreement set of a specific model. We define three types of utility metrics for the model:

- *Utility Precision* (U_P) Percentage of cases that are manually verified as true errors in the disagreement set of the model.

$$U_P = \frac{E_M \cap E_T}{E_M} \tag{8}$$

- *Utility Recall* (U_R) Percentage of true errors that occur in the disagreement set of the model among all the manually identified true errors.

$$U_R = \frac{E_M \cap E_T}{E_T} \tag{9}$$

- *Utility F-score* (U_F) Harmonic mean of utility precision and utility recall.

$$U_F = \frac{2 \times U_P \times U_R}{U_P + U_R} \tag{10}$$

4.4 Feature Selection

We used random forest for feature selection to compare against our proposed method of choosing features. Feature selection using random forest has been shown in [10] to demonstrate the most optimal results compared to other methods of feature selection. Because the memory requirement to train the random forest on the complete dataset with all features is too large, we sampled a subset of the dataset instead. After training the random forest with this sampled training set, we extracted the feature importance values generated by the model. We compared our 3458 features with the top 3458 and 8000 features (based on the feature importance), and all features using both the complete and partial datasets. Table 5 shows the accuracies

Table 5 Feature selection accuracies achieved by our models

System	3458 (Proposed)	3458 (RF)	8000 (RF)	All
SVM (sampled)	0.631	0.418	0.471	**0.675**
SVM (full)	0.675	0.473	0.520	**0.684**
RF (sampled)	**0.605**	0.463	0.478	0.569

The bold numbers signify being the relatively highest among their comparisons

Table 6 Utility for identifying errors

System	U_P	U_R	U_F
SVM (3K)	0.66	0.68	0.67
SVM (All)	0.65	0.64	0.64
RF (3K)	**0.69**	**0.78**	**0.73**

The bold numbers signify being the relatively highest among their comparisons

Table 7 Distribution of errors detected by each model

System	Error type	Count (%)
SVM (3K)	Incorrect name extraction	45 (21.43%)
	Incorrect name normalization	67 (31.90%)
	Incorrect KB attribute	98 (**46.67%**)
SVM (All)	Incorrect name extraction	40 (20.10%)
	Incorrect name normalization	68 (34.17%)
	Incorrect KB attribute	91 (**45.73%**)
RF (3K)	Incorrect name extraction	52 (21.49%)
	Incorrect name normalization	89 (**36.78%**)
	Incorrect KB attribute	101 (41.74%)

The bold numbers signify being the relatively highest among their comparisons

achieved by each system, which are computed based on the legacy labels. We can see that for all systems, our proposed feature selection method outperforms the random forest feature selection for both 3458 as well as 8000 features. Since SVM shows the best performance when using all features, we include this method, i.e., SVM (All), in our comparison against SVM and random forest with 3458 features, i.e., SVM (3K) and RF (3K), respectively, in our experiment.

4.5 Results

The utility values of our models for identifying errors in the legacy labels are shown in Table 6. We see that random forest performs better than SVM in terms of all the utility scores (achieving utility precision 0.69, utility recall 0.78, and utility f-score 0.73) and thus is more effective in identifying errors in the legacy labels of employer industry.

To better understand the reason, we checked the distribution of each type of errors detected by each model, which is shown in Table 7. It can be seen that SVM detected a larger ratio of KB errors, whereas random forest detected a larger ratio of normalization errors. Note that a normalization error often indicates that jobs from several different wrong employers (maybe different industries) are aggregated together, whereas KB errors mean that jobs from only the single correct employer (single industry) are aggregated. Therefore, the feature vectors associated with normalization errors can be much more complex and mixed than the feature vectors associated with KB errors. An example of a normalization error we encountered is for the Normalized Name of "Protech Corporation," the associated Unnormalized Names include "Protech Solutions" and "Pro-tech Search, Inc," in addition to "Protech Corporation" itself. "Protech Corporation" is a specialty construction company (sector 23), whereas "Protech Solutions" and "Pro-tech Search, Inc." are information technology services (sector 51) and staffing companies (sector 56), respectively. Thus, in this case, the feature vectors we derived for this particular employer ended up containing signals from three different industries. The results in Tables 6 and 7 show that random forest is better at capturing the interactions between different signals, especially in more complex and mixed feature vectors. Random forest can accommodate these relations, while SVM is unable to use the combinations so well. It is also shown in [18] that ensembles methods such as boosting are better able to capture the complex interactions between features, which is a potential advantage over SVM.

Next, we compare the error detection utility scores of our models between different industries. Figure 2, 3, and 4 show the results. Consistent with the overall trend, random forest has the highest utility f-scores in most of the industries.

One question we had by looking at Fig. 2, 3 and 4 is whether different industries have different complexities. So we looked at the error rate per industry within the whole test set, which is shown in Fig. 5. As the trend of error rates exhibited is similar to that of the utility f-score values shown in Fig. 2, we computed the linear correlation between these two sets of numbers. The correlations for SVM (3K), SVM (All), and RF (3K) are 0.52, 0.20, and 0.57, respectively. This means that our models generally perform better when detecting errors for industries that have higher error rates. In particular, sector 48, which has the lowest error rate among all sectors, resulted in the lowest f-scores for the SVM models and also the fourth lowest f-score for the random forest model.

One particularly difficult industry to train on and predict is the combined sector 54 or 56 largely because of the types of companies that fall under this category. The utility metrics for this category tend to be relatively lower compared to the other sectors. Many jobs within sector 54 are related to professional services and consulting, which means this industry commonly overlaps with other sectors. For example, financial services under sector 54 can be categorized under the finance industry (sector 52), and human resources consulting under sector 54 can be grouped with the staffing industry (sector 56). Moreover, most of the jobs in the staffing industry (sector 56) are directly related to specific industries for which the staffing

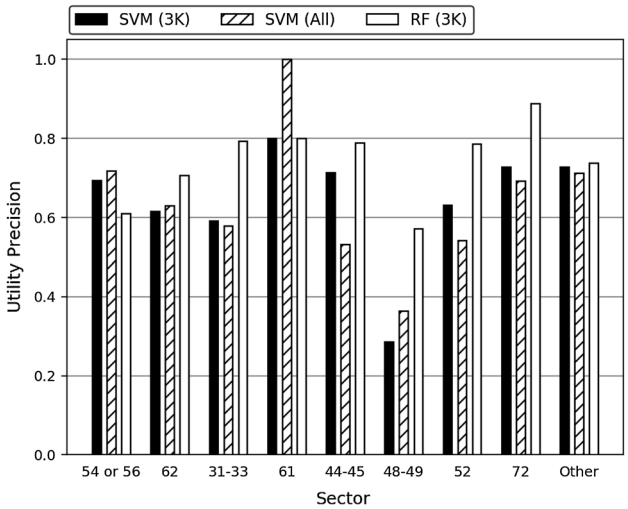

Fig. 2 Utility precision for identifying errors per industry

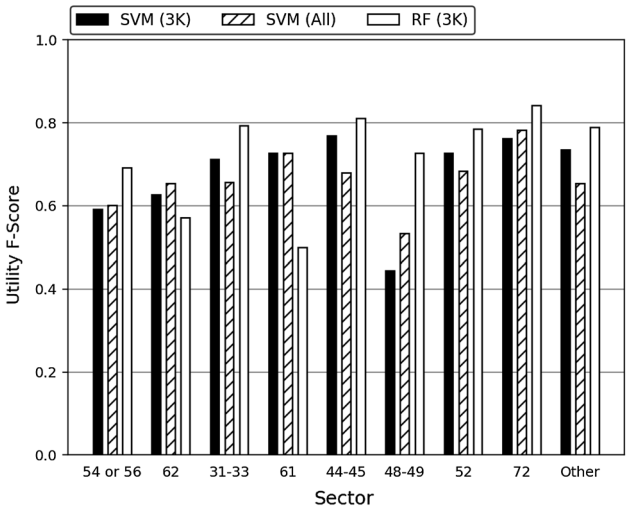

Fig. 4 Utility f-score for identifying errors per industry

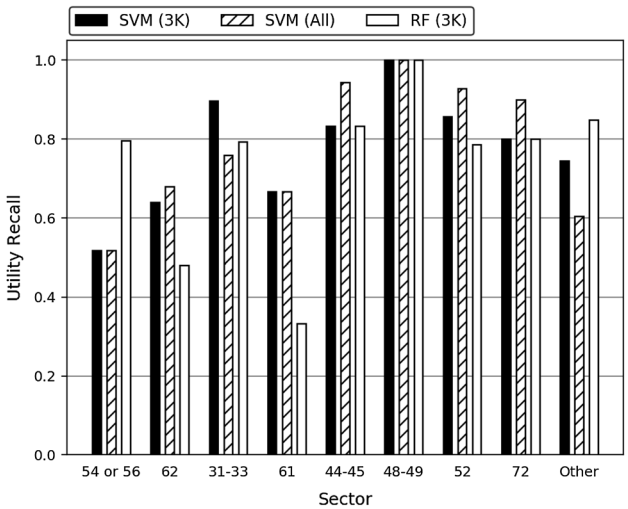

Fig. 3 Utility recall for identifying errors per industry

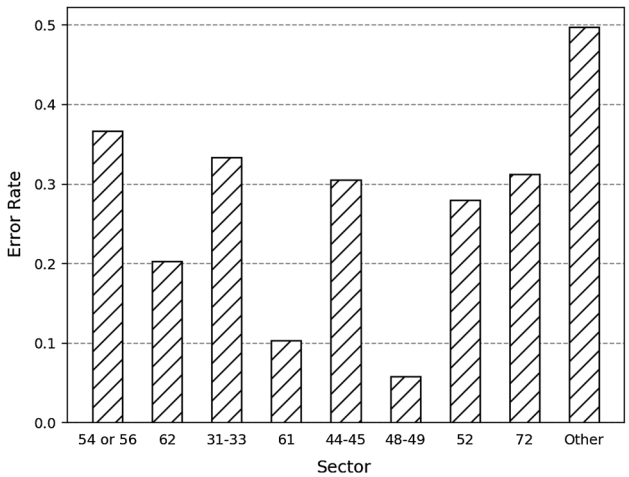

Fig. 5 Error rate per industry in test set

companies focus on and the jobs are generally very diverse. In this paper, we attempted to address this challenge by integrating the job title entropy and job description staffing rate to better identify such companies. However, according to our feature ablation experiment in Sect. 4.7, these two features exhibit very limited contributions.

4.6 Model Error Analysis

Model errors occur when the legacy system produces correct result, i.e., when the Unnormalized Name is valid, the normalization process is correct, and the industry attribute in the KB is marked correctly, but the model (SVM or random forest) predicts incorrect industry for the employer.

One source of model errors is that we assumed that the most frequent jobs posted by an employer are related to its industry; however, there are cases where the assumption does not hold true. For example, "Union Pacific" is a valid Unnormalized Name and is normalized to "Union Pacific Corporation," which is a transportation company, and is labeled correctly in the KB. However, the model predicts it as a non-transportation employer. The reason for this is that the employer name does not contain any keywords in the vocabulary for transportation industry, and moreover, the most frequent jobs posted by this employer are non-technical jobs like "Dispatcher", and "Track Laborer", which are not directly related to transportation industry.

Another source of model errors is that while we assume each employer has a single main industry in this paper, which can be reflected by the main jobs posted by this employer, there can be cases where the employer industry spans multiple sectors. For example, the company

Table 8 Results for different combinations of feature sets

Base	S1 Titles	S2 Company Keywords	S3 Employer desc. Keywords	S4 Job desc. Staffing rate	S5 Job title Entropy
∅	**0.603**	0.551	0.509	0.331	0.331
S1	N/A	**0.644**	0.638	0.603	0.619
S1, S2	N/A	N/A	**0.669**	0.641	0.649
S1, S2, S3	N/A	N/A	N/A	**0.673**	0.670
S1, S2, S3, S4	N/A	N/A	N/A	N/A	**0.675**

The bold numbers signify being the relatively highest among their comparisons

"Legoland" is a theme park chain (i.e., the recreation industry, sector 71) that hires for park personnel positions; however, there are also a large amount of food industry jobs within its postings because of the restaurants within its parks. As a result, while our manual label agrees with legacy label for this employer, which is sector 71 (belonging to the "Other" class), our model classified it as sector 72 (i.e., Accommodation and Food Services). Furthermore, conglomerate companies, which consist of multiple subsidiary companies, tend to span across multiple industries. "Ashley Furniture Industries" is an example conglomerate that owns the "Ashley HomeStore" and "Ashely Distribution Services, LTD" brands. These companies represent the manufacturing, retail, and transportation industries, which all get grouped under one entity. Therefore, such diversification makes it difficult to assign a specific industry to such employers and consequently makes it more difficult for the models to learn and predict the correct industry.

The last major cause of model errors that we observed is incorrect predictions caused by strong signals from another industry. For example, our model predicted sector 72 (Accommodation and Food Services) for "Target Corporation," which is an employer in Retail industry (sector 44–45). The main reason for this is that the keywords "guest" and "guests," which are very frequent within sector 72 employer descriptions, also happen to be very common within the Target Corporation description in our data. Similarly, the employer "Mvp Health Care, Inc." in Insurance industry (sector 52) was wrongly predicted as sector 62 (Health Care and Social Assistance) mainly due to the "health" and "care" keywords in the company name.

4.7 Feature Ablation

We performed experiments by running different combinations of features with SVM and comparing the accuracy to know the effect of each type of features in predicting the employer industry. The accuracy value achieved by each model here is also calculated based on the legacy labels. The results are shown in Table 8.

We can see that the normalized job titles alone capture a good amount of information about the employer industry, and they are able to capture more information compared to the rest of the features alone. Including both the company and employer description, keywords provide additional improvements in the results, over using just the normalized titles, for both SVM and random forest. The job description staffing rate and job title entropy do not provide much improvement in the result when included and are almost similar to using just the first three features. The reason may be that these two features are mainly designed for predicting the staffing industry (sector 56), and therefore, they may not be useful for predicting other sectors. On the other hand, the first three features may already cover the information captured by these two features for predicting the staffing industry.

5 Discussion and Future Work

Our method is based on the assumption that the most frequent jobs posted are related to the employer's industry. However, it might not hold true because the number of jobs and the types of jobs posted by an employer can change with time depending on the vacancies they have. To solve this problem, we wish to include additional features such as employer description (which can be obtained from the employer's Web site, e.g., the text under the "About Us" webpage, or from the Google Search Snippet). Note that although we incorporated employer description keywords as features, because our employer description data is extracted from job postings, certain descriptions may be inaccurate, missing, or lacking in useful information. As such, we believe extracting the employer descriptions directly from more reliable sources will improve the reliability of this feature set.

We also notice that there are certain employers that span across multiple industries (e.g., Apple, General Electric), also known as industrial conglomerates. There can be more than one industry label for these employers, and this diversification makes it difficult to assign a specific industry to such employers. In this paper, we assume each

employer has a main industry, which can be reflected by the main jobs posted by this employer. As mentioned in [17], some people choose the main industry as the one that generates the maximum revenue for the employer. In the future, the classification for such employers could be accomplished using a multilabel classifier in which an employer can be assigned multiple class labels. We could also consider this as a fuzzy classification problem.

As another future work, we plan to set up a semiautomatic feedback loop for our legacy industry classification based on the machine learning models developed in this paper to continuously detect errors in our legacy labels. Following each iteration of error detection, the employers that are detected by the machine learning models to be potentially erroneous can be sent off to our quality assurance (QA) team to manually verified and if necessary corrected. We then update our extraction process, normalization process, and KB attributes based on the manual corrections and restart the loop again. Each iteration after the detected errors are corrected, our models will generate a new disagreement set, but the utility performance will likely drop because as the easier cases get fixed, the remaining errors become more difficult to catch.

Finally, we plan to expand our models to learn and predict on all 20 industry sectors by separating out the combined "54 or 56" and the "Other" categories into individual sectors. This way, we can extract the features specific to each industry, which may allow us to more precisely pinpoint the errors that lie within these sectors. In particular for sectors 54 or 56, where there exist employers with overlapping industry classification, we will investigate more definitive features to potentially distinguish between them.

6 Related Work

Industry Classification Codes such as NAICS and SIC have been used for different analyses and applications. For example, the authors in [13] use different Industry classification codes to sample high-technology firms. [17] provides an overview of different industry classification codes and describes the challenges in assigning an industry to an employer.

In this paper, we used the industry sectors defined in NAICS 2017 [1] as our classes. We extended the work in [9] to incorporate all the industry sectors, including the staffing industry. We used multiclass classifiers, instead of the individual per industry binary classifiers, to learn and predict a total of nine classes. In addition to the normalized job title and company name keyword features, we generated an extra set of features consisting of employer description keywords to better distinguish between sectors. Finally, more comprehensive experiments including feature selection and per industry analysis have been performed in this paper.

A relational vector space model for entity classification is presented in [2], where each entity is represented as a vector of weights. The authors try to identify the group membership of entities linked by a particular relationship, with the assumption that linked employers often have the same affiliation. They use company industry classification as an example application to demonstrate the effectiveness of the proposed methods. The SIC (Standard Industry Classification) codes are used to classify the company industries. Two companies are linked if they co-occur in a business news story, or if they have other relationships, e.g., joint ventures, mergers/acquisitions, product/market related, and so on. Our work used a different type of data, i.e., job postings, to learn the signals indicating employer industry, which is orthogonal to their approach.

The authors in [19] show that SVMs perform reasonably well on the task of multiclass text classification and that one-versus-all classification performs favorably as compared to other approaches. [22] proposes an improved random forest algorithm for classifying high-dimensional text data with multiple classes. The proposed algorithm uses a new feature weighting method for subspace sampling and tree selection, and thus, the subspace size is reduced and classification performance is improved. The work in [20] handles the problem of class imbalance in data by reducing cost-sensitive classification to one-sided regression. The approach can be viewed as an extension to the one-versus-all SVM, which is based on estimating the components of the cost vectors directly via regression, and uses a regression loss function that reflects the cost of interest. The authors show that the algorithm performs better than one-versus-all SVM and many other existing SVM-based cost-sensitive classification algorithms. For our SVM models, we used the one-versus-one approach available in LIBSVM [6] to achieve the multiclass classification, which trains a single binary classifier for each pair of label categories. As such, the performance may still be impacted by the class imbalance. Hence, in the future, we can try the approach in [20] to perform cost-sensitive classification of employer industries on the job data in order to better overcome the problem of class imbalance.

A comparison of different models for supervised learning in [5] shows that ensemble methods such as boosted decision trees give the best average performance, and SVMs perform nearly as well as boosted trees. The authors in [18] also compare stochastic gradient boosting and SVM for predicting genomic breeding values. It is shown that boosting performs a little better than SVM since boosting is better able to capture the complex relations between features. In our experiments, we found that random forest performed better than SVM in detecting errors in our

legacy employer industry classification system by better capturing the interactions between different signals, especially in more complex and mixed feature vectors.

7 Conclusion

It is important to know the industry of an employer to get an insight about the demand in each industry. CareerBuilder uses an employer KB and an employer name normalization system to normalize different mentions of an employer to an entity in the KB. However, errors can occur in the normalization system, and the KB too can have incorrect industry attributes. Thus, we might get a wrong employer industry this way. Also, since the KB is huge with about 20M entities, it is not possible to manually detect the errors. Hence, we used machine learning models to infer the employer industry and automatically detect errors. We found that the job posting data, including job titles, employer names, and employer descriptions are useful indicators of the employer industry and hence used these as features for the models. We built multiclass classifiers for nine industry classes using two machine learning models: support vector machine (SVM) and random forest. Our experiments showed that random forest is more effective than SVM in identifying the errors in the existing industry classification system, which achieves precision 0.69, recall 0.78, and f-score 0.73. It especially better captures the interactions between different signals in more complex and mixed feature vectors, when normalization errors occur. We also observed that generally for industries that have higher error rates, our models perform better in detecting errors, according to a moderate correlation between the f-scores and the error rates for different industries. We demonstrated that job postings can be used to automatically detect errors in the existing system at CareerBuilder.

References

1. 2017 North American Industry Classification System (NAICS). https://www.census.gov/cgi-bin/sssd/naics/naicsrch?chart=2017. Accessed 12 Jan 2018
2. Bernstein A, Clearwater S, Provost F (2003) The relational vector-space model and industry classification. In: In proceedings of IJCAI workshop on statistical models from relational data, pp 8–18
3. Bhagat RC, Patil SS (2015) Enhanced smote algorithm for classification of imbalanced big-data using random forest. In: 2015 IEEE international advance computing conference (IACC). IEEE, pp 403–408
4. Breiman L (2002) Manual on setting up, using, and understanding random forests v3. 1. Statistics Department University of California Berkeley, CA, USA 1
5. Caruana R, Niculescu-Mizil A (2006) An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd international conference on machine learning. ACM, pp 161–168
6. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm. Accessed 11 Dec 2017
7. Dittman DJ, Khoshgoftaar TM, Napolitano A (2016) Is data sampling required when using random forest for classification on imbalanced bioinformatics data? In: Theoretical information reuse and integration. Springer, pp 157–171
8. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
9. Goindani M, Liu Q, Chao J, Jijkoun V (2017) Employer industry classification using job postings. In: IEEE international conference on data mining workshops, ICDM Workshops
10. Gromski PS, Xu Y, Correa E, Ellis DI, Turner ML, Goodacre R (2014) A comparative investigation of modern feature selection and classification approaches for the analysis of mass spectrometry data. Analytica chimica acta 829:1–8
11. Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer series in statistics. Springer, Berlin
12. Javed F, Luo Q, McNair M, Jacob F, Zhao M, Kang TS (2015) Carotene: a job title classification system for the online recruitment domain. In: Proceedings of the 2015 IEEE first international conference on big data computing service and applications, BIGDATASERVICE '15, pp 286–293
13. Kile CO, Phillips ME (2009) Using industry classification codes to sample high-technology firms: analysis and recommendations. J Account Audit Financ 24(1):35–58
14. Liu Q, Chao J, Mahoney T, Chern A, Min C, Javed F, Jijkoun V (2018) Lessons learned from developing and deploying a large-scale employer name normalization system for online recruitment. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, KDD '18
15. Liu Q, Javed F, Dave VS, Joshi A (2017) Supporting employer name normalization at both entity and cluster level. In: KDD, pp 1883–1892
16. Liu Q, Javed F, Mcnair M (2016) Companydepot: employer name normalization in the online recruitment industry. In: KDD, pp 521–530
17. Lyocsa S, Vyrost T (2009) Industry classification: review, hurdles and methodologies. SSRN. https://ssrn.com/abstract=1480563. Accessed 08 Jan 2018
18. Ogutu JO, Piepho HP, Schulz-Streeck T (2011) A comparison of random forests, boosting and support vector machines for genomic selection. In: BMC proceedings. BioMed Central, vol 5, p S11
19. Rennie JD, Rifkin R (2001) Improving multiclass text classification with the support vector machine. Tech. rep., Artificial Intelligence Laboratory, Massachusetts Institute of Technology. http://people.csail.mit.edu/jrennie/papers/aimemo2001.pdf. Accessed 10 Jan 2018
20. Tu HH, Lin HT (2010) One-sided support vector regression for multiclass cost-sensitive classification. In: ICML, pp 1095–1102
21. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
22. Xu B, Guo X, Ye Y, Cheng J (2012) An improved random forest classifier for text categorization. JCP 7(12):2913–2920