# Artificial Neural Networks Applied to Named Entity Recognition of Structured Data Sets

Master's Thesis, Submitted on 10/23/2017

Miriam Herman
Yeshiva University - Department of Mathematical Sciences
215 Lexington Ave
New York, NY
miriam.herman@mail.yu.edu

## ABSTRACT

This paper demonstrates the utilization of Artificial Neural Networks (ANNs) to classify the contents of column data in structured data sets. Using 100 dimensional pre-trained word embeddings from GloVE, we demonstrate a significant improvement over the existing Stanford NLP, OpenNLP, and NLTK toolkits for the task of classifying names, organizations, and addresses. We also introduce a classifier for products. Our best model has 99.74 % percent accuracy on unseen sets. We also use LIME to demonstrate the viability of our models and fine tune the model for handling of data classified incorrectly.

## Keywords

Artifical Neural Networks; ANN; Named Entity Recognition;NER; structured data

## 1. INTRODUCTION

According to a survey conducted by CrowdFlower, a platform for data scientists, "data preparation accounts for about 51% of the work of data scientists" and "60 % of data scientists view data preparation [collecting, labeling, cleaning, and organizing data] as the least enjoyable part of their work." [1] The report also stated that 49 % of a data scientists work involves structured data. It would therefore be extremely valuable to provide a rigorous method for tagging the contents of structured data sets to minimize the amount of time data scientists must spend on the tasks they enjoy the least. There are many other use cases for a classifier of structured data sets.

We compare the results of our state of the art ANN models to baselines from Stanford NLP, openNLP, and Python's NLTK toolkit to find a 99.7 % accuracy improvement in the best case for Neural Networks and a 16% improvement in the worst. Across all models, the average improvement is 75% with the Neural Network model. Our source code and training sets can be found at https://github.com/miriamherm/ClientClassification.


DOI: 10.1145/1235

## 2. DATA, BASELINES, AND EXPERIMENTS

In this section I share details of the data collection, the results of the baseline classifiers with examples, and the architecture for a Neural Network that achieves near perfect accuracy on the test data set. I then show the Neural Network is making predictions based on relevant information in the training data, and present a viable way to fine-tune the model for error correction.

### 2.1 Data Collection and Cleansing

Data for training and testing was acquired from a variety of sources, with duplicates removed. The appendix contains a complete list of data sources used, with notes regarding file name, column name and any additional steps taken on the data.

Once all the data was collected, the data was divided by type and compiled for each type into a single file. All duplicates were removed. The final count includes 723,553 unique addresses, 1,061,544 unique companies, 249,227 unique peoples names and 559,591 unique products. The data sets were then shuffled, and 20% of each set was removed for testing purposes.

We also generated a set of 250,000 names using a datafactory [3] that utilizes US census information to create name data determined by name frequency. We perform tests and training on names data with and without the generated names.

### 2.2 Baselines

In order to assess the necessity of a Neural Network trained classifiers for structured data, three baseline measures were tested; Stanford NLP NER [4], OpenNLP, and Python's NLTK package.

We tested the models on address, company, and names data, using the following metric. A tag was labeled correct if all words in a cell had the same label, and that label was correct, otherwise it was labeled incorrect. If the model allowed, we also show the metric for all words in a cell having at least once the correct label, and the rest "other."

The results of each model are below.

#### *2.2.1 Stanford NLP NER*

We used the Stanford NLP NER tool [4] to classify addresses, organizations and people, but as right from the box

it is only capable of classifying 7 specific types: Location, Person, Organization, Money, Percent, Date, Time, we did not use it to classify product data.

Using the above first metric, we found the Stanford NLP could not classify addresses. It wasn't able to classify addresses because it classified items crucial to identifying addresses, such as street numbers and words like "Rd" and "Ave" as "Other". A typical example of classifier output is: 603/O HINMAN/PERSON RD/O where "603" and "RD" were labeled as "Other" and "FINMAN" labeled as "PERSON". For the 144,709 addresses in the test set, the model had 0% accuracy, labeling only 2 address correctly. If we consider the metric of accuracy to be "correct label" and "other" the accuracy on this set increases to 9.28% or 13,431 addresses labeled correctly.

We also found the Stanford NLP NER classifier struggled to classify organizations for a very similar reason the model failed to classify addresses: words crucial to an organization, such as "Limited", are classified as "Other". A typical example of the output for an organization cell is: ALASTAIR/PERSON WRAY/PERSON LIMITED/O. Since many companies are named after their founder, they are often confused for names. For the 212,307 companies in the test set, the model had 0.018 % accuracy, labeling only 38 companies correctly.If we consider the metric of accuracy to be "correct label" and "other" the accuracy on this set increases to 8.95% or 19,009 companies labeled correctly.

Since the Stanford NLP NER classifier does not distinguish a single letter as a name, and because a lot of our name data consists of first and middle names listed with just an initial, the Stanford NLP NER struggled to classify people as well. A typical example of the output for a name is:I/O SONI/PERSON, where the initial is labeled as "Other". Of the 49,844 names in the test set, the model had 3.66 % accuracy, labeling 1825 examples correctly. If we consider the metric of accuracy to be "correct label" and "other" the accuracy on this set increases to 52.73% or 26,287 names labeled correctly.

When the Stanford NLP NER classifier was tested on the collected names data together with the datafactory generated names data, it performed significantly better, labeling 39.69% or 35,634 out of 89,789 names correctly. (Missing 2486 generated names.) If we consider the metric of accuracy to be "correct label" and "other" the accuracy on this set increases to 71.89% or 64,548 names labeled correctly.

Out of 406,860 examples, only 1865 or 0.5% of examples were labeled correctly.

### 2.2.2 OpenNLP

We used the OpenNLP NER [2] tool to classify addresses, organizations and people, but like StanfordNLP, it is only capable of classifying; Location, Person, Organization, Money, Percent, Date, Time right from the box. We did not attempt to train it to classify product data.

OpenNLP performed much better than Stanford NLP when classifying addresses because it seems to recognize certain words common to addresses, like "Ave" and "St". However, it does not recognize other crucial words, like "Rd"s and misclassified most "Rd"s as "Organizations". Out of 144,709 addresses in the test set, OpenNLP classified 19.22% or 27,818 examples correctly.

This model also performed much better than Stanford NLP when classifying companies, recognizing common company abbreviations like "LTD" and "CO" as part of an organization. However, it failed to recognize common company words like "Limited" and "Agency". Out of 212,307 companies in the test set, OpenNLP classified 36.39% or 77,265 examples correctly.

When classifying names OpenNLP performed significantly better than Stanford NLP because it classified full terms, and didn't classify names with initials if the last name was recognized strongly as a name. A name like "A G QUITO," with an uncommon last name, would have been misclassified as an organization and some names that were not formatted well were left unclassified, such as "B HARRIS". Out of 49,844 names in the test set, OpenNLP classified 63.16% or 31,483 examples correctly.

When the OpenNLP NER classifier was tested on the collected names data together with the datafactory generated names data, it performed slightly better, labeling 70.58% or 63,374 out of 89,789 names correctly.

Out of 406,860 examples OpenNLP classified a total of 136,576 or 33.6% examples correctly.

### 2.2.3 NLTK Classifer

We attempted to classify addresses, organizations and people using the Natural Language Toolkit (NLTK) [6]package in Python. The NLTK package can classify; Organization, Person, Location (for example Mount Everest), Date, Time, Money, Percent, Facility, GPE (Geo-political entities, like city, state/province, country). We used the toolkit straight from the box, using it to classify by Organization, Person, and GPE and did not attempt to train it to classify product data.1

NLTK performed much better than Stanford NLP when classifying addresses but worse than OpenNLP. It seems to recognize certain words common to addresses, like "East" and "Indian" but not the crucial words like "st", "rd", and "ave". The NLTK classifier misclassified addresses as organizations, people, and other, but very rarely confused them with several mixed labels. Out of 144709 address in the test set, NLTK classified 2.61% or 3,771 examples correctly.

This model performed much better than Stanford NLP and OpenNLP when classifying companies, recognizing common company abbreviations like "LTD" and "CO" as part of an organization. It is hard to understand why it classified some companies incorrectly. Out of 212,307 companies in the test set, NLTK classified 82.73% or 175,631 examples correctly.

When classifying names NLTK performed worse than Stanford NLP and OpenNLP, and it is unclear why. On occasion NLTK would correctly identify a name with initials as a person, and in other cases it would classify them as organizations, GPEs, or other. There does seem to be a correct trend of classifying names as "Person" if there is a first and middle initial, but this is not consistent. Out of 49,844 names in the test set, OpenNLP classified 3.46% or 1727 examples correctly.

When the NLTK classifier was tested on the collected names data together with the datafactory generated names data, it performed better, labeling 20.43% or 18,346 out of 89,789 names correctly.

Out of 406,860 examples OpenNLP classified a total of 181,129 or 44.52% examples correctly. (This increase in total accuracy is due to NLTK's proficiency at classifying companies.)

**Table 1: Stanford NLP Confusion Matrix**

| | Address | Company | Name | Product | Other | Mixed |
|---|---|---|---|---|---|---|
| Address | 0.000 | 0.000 | 0.000 | NA | .249 | .750 |
| Company | 0.000 | 0.000 | .002 | NA | .573 | .425 |
| Name | 0.000 | 0.000 | .037 | NA | .140 | .824 |
| Product | NA | NA | NA | NA | NA | NA |

**Table 2: OpenNLP Confusion Matrix**

| | Address | Company | Name | Product | Unclassified | Mixed |
|---|---|---|---|---|---|---|
| Address | .192 | .257 | .040 | NA | .076 | .435 |
| Company | .001 | .364 | .003 | NA | .002 | .630 |
| Name | .003 | .012 | .632 | NA | .061 | .292 |
| Product | NA | NA | NA | NA | NA | NA |

## 2.3 Neural Network Training

In my previous work a thorough analysis of embeddings, including; Facebook fastText, word2vec, continuous bag of words dependency based embeddings, character embeddings, and more was done. We learned that one of the best pre-trained embeddings for this task are the 100 dimensional GloVe word vectors. Our models were trained using this embeddings.

We set out intending to perform a grid search for the parameters of our model. We began our tests with the simplest model and created binary models for each data type. Each model had a single layer and 128 nodes, was optimized with adam and trained with 10 epochs. The four models achieved 97%-99% accuracy. As this was huge improvement from all the previous baselines, and because no model will have 100% accuracy, we did not see a need to fine tune the architecture further.

Our models were trained on 2,016,028 unique phrases containing 389,808 unique words. The max number of words in a phrase was decided to be 10, so our model was trained on 10 dimensional word vectors. If a phrase had fewer than 10 words, it's associated vector was padded with zeros. We trained two models, one with the datafactory generated names and one without. Both models were subsequently tested on our test set and on a generated business data set. The models generated probability predictions for the each type. The results are presented.

### *2.3.1 Test Set Results*

We trained two models, one with generated names and one without, and when tested on the test set, for all the data types aside from names, the two models performed almost identically. For names, the model trained with the generated names performed 4% better, from 94% to 98%, and therefore going forward in this paper we will train models on the generated names. The address classifier hovered around 99%, the companies classifier around 98% and the product classifier at 96%.

## 2.4 LIME - Explaining the Predictions

Machine learning algorithms are black box solutions, where input is fed to a model and a classification returned without explanation as to how the decision was made. LIME, LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS [5], attempts to solve this problem by perturbing data, running the model on the perturbed data, and seeing the classification result. In this way it is able to discern which words were most crucial to the classification process.

We provided LIME with 100 words from each type and list below the top 6 results. A complete list results can be found in the appendix.

As was hoped for an address classification model, Ave, Rd, and St are among the 6 most important words. For the company classifier we are please to see Limited and Property at the forefront, but are surprised That ACTUARIAL and CJH make the top of the list, however this may be due to only providing 100 phrases to the model. Unsurprisingly, very popular last names make the top of the names list, and the results of the Product test explain why addresses and dates were being mislabeled as products: numbers and mixtures of numbers and letters, like "500" and "2B", carry a lot of weight in the classification of a product.

## 2.5 Business Set Results

In order to validate the accuracy of the above Neural Networks results, we tested the models on a business data set kept completely separate from the data collection process. The vanilla network performed well on names, companies, and obviously identified address parts, however it struggled to classify addresses without suffixes and products.

The model especially struggled classifying dates, labeling them as products with almost 100% confidence. Also, due to single letter initials in the names data, entries with single letters, like "pay cycle" were classified as a name with 98% likelihood.

Due to the LIME results above we understand some of the difficulties the model had, City, Street, and Country have none of the 6 most important words for an Address. Dates are mixtures of letters and numbers, much like products. We trained an LSTM model on 1 and 5 epochs to see if a longer memory network could pick up the patterns with names and products that the vanilla network missed.

Unfortunately the LSTM models did not show improved performance. Its results are below.

### *2.5.1 LSTM*

We ran the model allowing 80000 features, 10 dimensional word vectors, 128 nodes, dropout and recurrent dropout of .2, and sigmoid activation. We trained for 1 epoch. It took more than 10X longer to run the single epoch that the previous model took for 10 epochs. Testing it took significantly longer as well, and the results were not great.

**Table 3: NLTK Confusion Matrix**

| | Address | Company | Name | Product | Other | Mixed |
|---|---|---|---|---|---|---|
| Address | .026 | .208 | .312 | NA | .449 | .005 |
| Company | .003 | .827 | .005 | NA | .094 | .070 |
| Name | .002 | .145 | .035 | NA | .792 | .026 |
| Product | NA | NA | NA | NA | NA | NA |

**Table 4: Generated Names NN Confusion Matrix**

| | Address | Company | Name | Product |
|---|---|---|---|---|
| Address | .997448 | .0017542 | .0003253 | .003252 |
| Company | .00128143 | .987642 | .002799 | .007761 |
| Name | .0010614 | .008826 | .989925 | .01163 |
| Product | .00407549 | .0178114 | .006845 | .965432 |

**Table 5: Collected Names NN Confusion Matrix**

| | Address | Company | Name | Product |
|---|---|---|---|---|
| Address | .996454 | .001618 | .000358 | .00376 |
| Company | .000656 | .987531 | .002567 | .007580 |
| Name | .006821 | .0619743 | .949441 | .0271665 |
| Product | .00291918 | .0184617 | .0068049 | .965063 |

The model only classified 79% of addresses correctly and 90% of names correctly. Although it did show an improvement on the test set for Products, up 2% to 98%, it made the same mistakes on the business data set as the vanilla network.

In order to determine the effects of additional epochs on this problem, I trained a binary model for products using the LSTM architecture above, but for 5 epochs. The results are below.

## 2.6 Fine-Tuning

Our model performed quite well on the data types it had seen before, such as names and companies, but was confused by some new data types like dates. Instead of retraining the entire model with the new data types, we fine tune the model with 8487 dates of different formats.

We fine-tune the model using Stochastic Gradient Descent. Freezing all layers except the topmost, we set the learning rate to 0.0001, decay to 1e-6 and momentum to 0.9 [7].

We compile and train each model with the new optimizer on 50 epochs, using only the negative examples.

### *2.6.1 Vanilla Network*

The runtime is less than a minute and accuracy on the product test set drops from 96.54% to 92.29% . However, the model initially classified dates as products with 89.20% confidence, and after the fine tuning, that confidence drops to 4.40%.

When we run the model on the business data set we find dates are classified as products with only 33.51%-43.10% confidence. We have achieved our goal, fine tuning the model to not mistake dates for products.

If we could improve the accuracy on the test set higher than 92.29% that would be ideal.

We lower the number of epochs in half, to 25, and find a 1% improvement to 93.44% accuracy on the test set, but that dates are classified as products in the business data with 74.60%-79.41% confidence. We lower the decay to 1e-4, as is written some examples, and the accuracy on the test set increases less than 1% to 92.54% to but dates are classified as products with 43.30-50.58% accuracy.

We leave further testing of fine tuning methods to the reader as the first method presented provides reasonable results for our task.

### *2.6.2 LSTM-1 epoch*

The runtime is close to 5 minutes and accuracy on the product test set drops from 98.35% to 98.23% . However, the model initially classified dates as products with 86.35% confidence, and after the fine tuning, that confidence barely drops, landing at to 84.09%.

### *2.6.3 LSTM-5 epoch*

The runtime is close to 10 minutes and accuracy on the product test set drops from 98.51% to 98.43% . However, the model initially classified dates as products with 86.88% confidence, and after the fine tuning, that confidence barely drops, landing at to 84.28%.

### *2.6.4 Missing Words*

Of the 389917 unique tokens found by the tokenizer, a total of 145,573 words (37%) were discluded from the embedding matrix; 8724 of 21,711 address words (40%), 35985 of 252378 company words (14%), 33669 of 66475 names words (51%), and 67195 of 106293 product words (63%). About 6% of the total number of texts.

Of the missing address words, the vast majority are numbers, including 981764 and 4615 and numbers with words like u52 and 569a. However LIME did list numbers very highly in its results for an addresses most important words.

Of the missing company words, the vast majority are non-English words, including kebabse, capitus, amav, and bevtex. Some words like limitmanagement were also discluded. Though it would seem most non-English words were discluded, LIME did list very unlikely words like CJF in its results for most important words.

Of the missing name words, every single one is an irregular name such as modrcin, repohl, and berdichevskaya.

Of the missing product words, many are numbers, dimensions, or non English words like colgadores, however there are some description words like fisherman's and minilight that are being ignored that may be causing the product classification difficulty.

A character embedding may be needed to resolve this problem.

# 3. CONCLUSIONS

In conclusion we have shown the viability of Neural Networks for learning the contents of structured data sets, and have proven that this method is an improvement on the existing software. We have also presented a simple usable

**Table 6: LIME - Most Important Words**

| Data Type | 6 Important Words |
|---|---|
| Address | Saddle, AVE, Rd, Suffield, MADISON, St |
| Company | LIMITED, PROPERTY, ACTUARIAL, CJH, PLAYING, WORLD |
| Name | SINGH, Frank, GALLIMORE, Harding, Bush, Remirez |
| Product | 500, Coffee, Non-Stick, 2B, Wine, rigolo |

**Table 7: LSTM 1 epoch - Confusion Matrix**

| | Address | Company | Name | Product |
|---|---|---|---|---|
| Address | .79172 | .000653 | .000198 | .000914 |
| Company | .00724 | .99316 | .00173 | .00397 |
| Name | .0562 | .0373 | .90744 | .023328 |
| Product | .05414 | .0092 | .004249 | .983564 |

**Table 8: LSTM 5 epochs - Confusion Matrix**

| | Address | Company | Name | Product |
|---|---|---|---|---|
| Address | .998772 | .000653 | .000198 | .000594 |
| Company | .000338 | .99354 | .00206 | .00362 |
| Name | .000678 | .00407 | .990576 | .00397 |
| Product | .001436 | .00768 | .00551 | .98511 |

architecture for this task, as well as tools for fine-tuning a model with a small set of negative examples. The source code for this project can be found at https://github.com/miriamherm/ClientClassification along with our training and test datasets.

## 4. ACKNOWLEDGMENTS

I would like to acknowledge my advisor, Dr Kavitha Srinivas, for the time and dedication she spent making sure my project was the best it could be. As well I would like to thank the Yeshiva University Department of Mathematical Sciences for their rigorous program in mathematics.

## 5. REFERENCES

[1] 2017 data scientist report. crowdflower.com, 2017.

[2] A. O. D. Community. Apache opennlp developer documentation. https://opennlp.apache.org/docs/1.8.2/manual/opennlp.html, 2011,2017.

[3] A. Gibson. Generate test data with datafactory. http://www.andygibson.net/blog/tag/datafactory/, 2011.

[4] T. G. Jenny Rose Finkel and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05 Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. ACM, June 2005.

[5] S. S. Marco Tulio Ribeiro and C. Guestrin. âĂIJwhy should i trust you?âĂİ, explaining the predictions of any classifier. In *KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, August 2016.

[6] E. K. Steven Bird and E. Loper. Natural language processing with python- analyzing text with the natural language toolkit. http://www.nltk.org/book/.

[7] F. Xu. A comprehensive guide to fine=tuning deep learning models in keras (part i). https://flyyufelix.github.io/2016/10/03/fine-tuning-in-keras-part1.html, 2016.

# APPENDIX

## A. DATA SOURCES

## B. LIME FULL RESULTS

**Table 9: Business Data Results**

| Col Name | Example | Vanilla NN | LSTM 1 Epoch | LSTM 5 Epoch |
|---|---|---|---|---|
| Account Owner | Toni Gomez | Person - 0.998 | Person - 0.827 | Person - 0.999 |
| Billing Address | Suite 10049 | Address- .999 | Companies- .04 | Address - .69 |
| Billing Contact | Allen Hardin | Person- .999 | Person- .7897 | Person- .999 |
| Billing Email | jgoff@ma1l2u.org | Company - .826 | Company - .811 | Company - .956 |
| City | Helena | Person - .429 | Company - .308 | Person .356 |
| Conversion Date | Thu Apr 03 05:20:32 EDT 2014 | Product - .999 | Product - .999 | Product - .998 |
| Country | Slovakia | Product - .430 | Product - .430 | Product - .413 |
| Custom Metrics | code,text,room | Product - .584 | Product - .596 | Product - .564 |
| Org Name | Morgan Studios | Company - .864 | Company - .875 | Company - .877 |
| Parent Name | 'Ringgold Cafe' | Company - .861 | Company - .832 | Company - .852 |
| Pay Cycle | 'Q','T' | Person - .982 | Person - .993 | Person - .997 |
| Pay Method | 'Invoice' | Product - .472 | Product - .540 | Product - .542 |
| po_num | PO6793946273 | Company - .575 | Company - .459 | Product - .397 |
| State | 'in', 'so' | Product - .664 | Product - .661 | Product - .880 |
| Street | 'Mcconnel' | Person - .501 | Person - .322 | Person - .456 |
| Terms | 'Standard' | Product - .661 | Product - .670 | Product - .552 |
| Valid From | Thu Jul 30 05:20:32 EDT 2013 | Product - .999 | Product - .999 | Product - .999 |
| Valid To | Sat Mar 15 05:20:32 EDT 2014 | Product - .999 | Product - .999 Product - .998 | |

**Table 10: Data Sources**

| Source | Data Type | # records | Notes |
|---|---|---|---|
| openaddresses.io | Addresses | 707094 | Concatenated number and street from US Northeast |
| sec.gov /rules/other/4-460list.htm | Companies | 951 | Removed headers |
| qualityfrozenfoods.com | Products | 5615 | Downloads |
| crownprod.com/ | Products | 5615 | Downloads |
| ikea.com | Products | 2765 | copied all products from site catalog |
| product-open-data.com/ | Products | 551953 | GTIN table GTIN_NM column |
| product-open-data.com/ | Companies | 4151 | brand table brand_NM column |
| wordlab.com | Companies | 4924 | company-names-list Removed top 66 and bottom 33 rows |
| wikipedia | Companies | 688 | List_of_common _carrier_freight _railroads_in _the_United_States |
| wikipedi | Companies | 1648 | List_of_companies _of_the_United_States |
| wikipedia | Companies | 103 | List_of_department _stores_of_the _United_States |
| wikipedia | Companies | 112 | List_of_independent _bookstores _in_the_United_States all listed with"in city" |
| wikipedia | Companies | 404 | List_of_supermarket _chains _in_the_United_States split on '(' and '-' |
| wikipedia | Companies | 66 | List_of_United_States _clock_companies |
| wikipedia | Companies | 259 | List_of_United_States _insurance_companies |
| wikipedi | Companies | 502 | List_of_United_States _water_companies |
| census.gov | People | 5494 | 1990_census_namefiles Names generated with most common first names concatenated with most common last names |
| data.gov | Companies | 1139 | Active_Benefit _Companies Business Name |
| data.gov | People | 237260 | Civil_List Name |
| data.gov | Companies | 4362 | Consumer_Complaints Company |
| data.gov | Products | 18 | Consumer_Complaints Product |
| data.gov | Addresses | 2112 | FOIL_Report - Trade_Waste_All _Approved_or_Denied Mailing Office |

**Table 11: Data Sources part 2**

| Source | Data Type | # records | Notes |
|---|---|---|---|
| data.gov | Companies | 2151 | FOIL_Report - Trade_Waste_All _Approved_or_Denied - Trade Name |
| data.gov | Addresses | 369 | IDOL_2013 _Registered_Owner _Rideṡcsv Address |
| data.gov | Companies | 1100 | IDOL_2013 _Registered_Owner _Rideṡcsv Ownername and Manufacturer |
| data.gov | People | 367 | IDOL_2013 _Registered_Owner _Rideṡcsv ContactName |
| data.gov | Products | 1731 | IDOL_2013 _Registered_Owner _Rideṡcsv Ridename |
| data.gov | Companies | 1504 | Licensed _Insurance _Companies Company Name |
| data.gov | Addresses | 488 | Lobbying _Reporting _System _Maryland _Registered _Employers _List _ Address |
| data.gov | Companies | 444 | Lobbying _Reporting _System _Maryland _Registered _Employers _List _ Firm Name |
| data.gov | People | 652 | Lobbying _Reporting _System _Maryland _Registered _Employers _List _ Concatenate( First Name, Middle Name, Last Name) |
| data.gov | Companies | 184 | Lobbyist _Activity _Contacts Lobbyist Firm |
| data.gov | Addresses | 270 | Lobbyist _Activity _Contacts_ Lobbyist |
| data.gov | Addresses | 5031 | M_WBE_LBE _and_EBE _Certified_Business_List-Address1 |
| data.gov | People | 5327 | M_WBE_LBE _and_EBE _Certified_Business_List-Contact_Name |
| data.gov | Companies | 5410 | M_WBE_LBE _and_EBE _Certified_Business_List-Vendor_Formal _Name |
| data.gov | Addresses | 200 | Neighborhood _and_Rural _Preservation _Companies _Directory Street Address |

**Table 12: Data Sources part 3**

| Source | Data Type | # records | Notes |
|---|---|---|---|
| data.gov | Companies | 205 | Neighborhood _and_Rural _Preservation _Companies _Directory Organization Name |
| data.gov | Products | 2306 | nsn _extract _4518-Common_Name |
| data.gov | Addresses | 7320 | Oregon _Consumer _Complaints Address 1 |
| data.gov | Addresses | 1038 | Oregon _Consumer _Complaints Address 2 |
| data.gov | Companies | 9403 | Oregon _Consumer _Complaints |
| data.gov | Companies | 38 | OWEB _Small _Grant _Teams Team |
| data.gov | People | 38 | OWEB _Small _Grant _Teams Team Contact |
| data.gov | Companies | 1798 | Prequalified _Firms - Prequalified Vendor Name |
| data.gov | Companies | 266 | SCA _Disqualified _Firms Vendor Name |
| data.gov | Companies | 47 | Science _Festival _Company _Sponsors Company Sponsor |
| data.gov | Companies | 58 | Top _50 _Employers _ _Hawaii _County - Name |
| data.gov | People | 54 | Top _50 _Employers _ _Hawaii _County Concatenate (Contact First Name, Contact Last Name) |
| hline data.gov | Addresses | 135 | Top _Manufacturing _Companies _in _SSMA _Region Primary Address |
| data.gov | Companies | 202 | Top _Manufacturing _Companies _in _SSMA _Region Company Name and Ultimate Parent |
| data.gov | People | 233 | Top _Manufacturing _Companies _in _SSMA _Region First Name and Last Name |
| data.gov | Addresses | 71 | Trade _Waste _Broker _Registrants Address |
| data.gov | Companies | 71 | Trade _Waste _Broker _Registrants Account Name |
| DBpedia | Companies | 82838 | Column B |
| DBpedia | People | 200946 | |
| data.gov.uk | Companies | 1,045,333 | BasicCompanyData |

**Table 13: LIME Results - Addresses**

| 100 words |
|---|
| ('Colony', 0.011473951103404855) |
| ('Rock', 0.0089593689938616706) |
| ('ST', -0.0088593440869146143) |
| ('472', -0.0087149165447808329) |
| ('Attawanhood', -0.0084193680321162229) |
| ('Trl', 0.0083109071638142514) |
| ('Crse', 0.0073590555140528148) |
| ('Ln', 0.0068177014744160123) |
| ('1173', 0.0067862905088006738) |
| ('St', 0.0066825214955059282) |
| ('Holcomb', 0.0066454675118291593) |
| ('MILFORD', 0.0065828182899055456) |
| ('Dr', 0.0062634731443634893) |
| ('Sharon', -0.0062604222173652593) |
| ('Three', 0.0060686596157848551) |
| ('70', 0.005944576885357245) |
| ('South', -0.0057737948566826316) |
| ('Cutlery', -0.0055069602681032583) |
| ('281', 0.0054479898148037725) |
| ('750', 0.0053575837734054482) |
| ('Fish', 0.0052517231180250834) |
| ('Ct', -0.005249845626465144) |
| ('Dziok', 0.005195532481860588) |
| ('MAIN', -0.0051827186689839761) |
| ('339', 0.0051711308051238377) |
| ('W', 0.0051267782330448094) |
| ('JARVIS', 0.0051169503896465887) |
| ('Long', 0.0051141868924031034) |
| ('Overlook', 0.0050983299215228445) |
| ('6', -0.0050572295784190825) |
| ('Mile', 0.0049800477202194141) |
| ('Pembroke', 0.0047896210377595562) |
| ('1226', -0.0047872522310372541) |
| ('67', -0.0046897959503926622) |
| ('Colonial', -0.0046597142972620072) |
| ('Hill', -0.0045926377398190525) |
| ('71', 0.0045063251439902627) |
| ('Pendleton', 0.0044695469127539848) |
| ('280', -0.0044146549356582545) |
| ('24', -0.0043828796785156656) |
| ('59', -0.0043713162217085119) |
| ('Middle', 0.0043517852493370618) |
| ('Pape', 0.0042792432102687471) |
| ('308', -0.0042628445800114226) |
| ('50', -0.0041895565069510702) |
| ('School', 0.0041876186438888421) |
| ('ALYCE', 0.0041780174901895799) |
| ('Ford', -0.0041424826686335221) |
| ('Mayflower', 0.0041262396914725826) |
| ('2768', 0.0040902863395010820) |
| ('US', -0.0040225689872512649) |
| ('AVE', 0.0039694894901525879) |
| ('S', 0.0039338102123051459) |
| ('PRATT', -0.0039145983503342287) |
| ('259', 0.0039065843632802169) |
| ('S', 0.0038705685903864871) |
| ('High', -0.0038649420880716490) |
| ('180', 0.0038492445991866468) |
| ('33', -0.0038243753658219843) |
| ('Willow', -0.0037921911871189738) |
| ('Gail', -0.0037642957560122398) |
| ('Porterbrook', 0.0037343963878524035) |
| ('150', 0.0037099380896738884) |
| ('Lakeside', 0.0036784420502833058) |
| ('Head', 0.0036616703009673783) |

**Table 14: LIME Results - Companies**

| 100 words |
|---|
| ('LIMITED', -0.6334913065099768) |
| ('BOWKER', 0.085768448232139916) |
| ('W', 0.085256963785598189) |
| ('A', 0.077019693074941054) |
| ('CROWTON', 0.063811590981113275) |
| ('LIMITED.I.C.', 0.063226970014220785) |
| ('BLUEBERRY', 0.047334262962764828) |
| ('VISION', 0.041637078213711549) |
| ('DIAMOND', -0.029686911972685942) |
| ('EMPLOYMENT', -0.028821850833759433) |
| ('SERVICES', -0.02675147127561733) |
| ('CHARIOT', -0.026343271954209862) |
| ('INNOVATIONS', 0.025761522824528798) |
| ('PROPERTY', -0.025056847403916681) |
| ('BOYLE', 0.024847613266834677) |
| ('PLUMBING', 0.023818811932263484) |
| ('CAPITAL', 0.023478350642488779) |
| ('GOURMET', 0.023280027593238065) |
| ('BURUNGA', 0.02179084173611746) |
| ('CCI', -0.021382209983426478) |
| ('PRIVATE', -0.021266271724758642) |
| ('CHESHIRE', -0.020814664069936307) |
| ('PRINT', -0.020803835364209447) |
| ('ALAMORT', -0.020644670993146316) |
| ('TRANS', 0.020573449048155883) |
| ('TRANSPORT', -0.019689917035936574) |
| ('PLUMBING,', -0.019673519729161787) |
| ('GIFTS', -0.019247381076709064) |
| ('MANAGEMENT', 0.019216652622711532) |
| ('LIGHTINTHEBOX', -0.018814686020860959) |
| ('ELECTRICAL', -0.01785203711354728) |
| ('COST', -0.017054545513881091) |
| ('ATOMIC', -0.01696395227116955) |
| ('OUTSKIRTS', -0.016135831769840563) |
| ('A.V.', -0.016044059045798663) |
| ('ASSIST', -0.01598204180976778) |
| ('AUTOMOTIVE', 0.015825302950403356) |
| ('RUTLAND', -0.015682944856849682) |
| ('APC', 0.015563790964872162) |
| ('CORPORATE', 0.015507539580536888) |
| ('PUBLISHING', 0.015501764794851217) |
| ('VEHICLE', -0.01521185070585581) |
| ('MITSUBISHI', 0.015166627019123965) |
| ('COCHRANE', -0.01514414968313427) |
| ('LIM', 0.015113258069838122) |
| ('CEILINGS', 0.014986838982052032) |
| ('CAVEAT', -0.014741092974705869) |
| ('SHIRT', -0.014316302671818684) |
| ('CELLWIZE', 0.014265175473772073) |
| ('LTD', 0.014226032630922888) |
| ('ALYASAMEEN', 0.014224222298585875) |
| ('(2017)', -0.01417865195848435) |
| ('ANDREI', -0.014154684427557516) |
| ('CDCE', 0.014066137159655703) |
| ('K', -0.013881831339935365) |
| ('PARTNERS', 0.013782206091035566) |
| ('LIMITEDMPANY', 0.013536027423489739) |
| ('LEARNING', 0.01352363395115847) |
| ('COVENTRYTED', -0.013435468891485898) |
| ('LTD.', 0.013396475386249987) |
| ('12', 0.013278325400936221) |
| ('CARDPRIZE', -0.013085957788976444) |
| ('DADDIES', 0.013035745507551596) |
| ('LTD"', -0.01281774107701851) |
| ('BUBBLE', -0.012725919749265787) |

**Table 15: LIME Results - Names**

| 100 words |
|---|
| ('Megan', 0.0020859877331014854) |
| ('MERCADO', 0.0020250263741921769) |
| ('Hannah', 0.0019930725456540452) |
| ('Kathleen', 0.0019790904760570 16) |
| ('Juanita', -0.0019744115520613763) |
| ('SAN', 0.0019040837049135831) |
| ('Mark', -0.0018448339205664032) |
| ('Nathan', -0.0017945064864189162) |
| ('Perkins', 0.0017878865465037453) |
| ('BUCHANAN', 0.0017517594348443204) |
| ('S', 0.0016787643173931241) |
| ('Moss', 0.0016702165905126752) |
| ('Shannon', -0.0016648960651855596) |
| ('DRAYCOTT', -0.0016103413375417037) |
| ('N', -0.0015932217856837851) |
| ('GOMEZDELATORRE', 0.0015674730881019994) |
| ('Tate', -0.0015607938369164806) |
| ('Carolyn', 0.0015489686799778567) |
| ('SANTIAGO', 0.0015378158818201404) |
| ('H', 0.0015279213568687488) |
| ('SANTANGELO', -0.0015166661825864027) |
| ('Morton', 0.0015151727001957659) |
| ('Anita', -0.0014943811064093021) |
| ('SCHILLING', 0.0014885028875234671) |
| ('LARRY', -0.0014770034154076221) |
| ('Heath', 0.0014724073394996673) |
| ('Roger', -0.0014635410620008019) |
| ('JIMENEZ', 0.0014430361931297168) |
| ('Valenzuela', 0.0014348824833932696) |
| ('A', 0.0014225054601604851) |
| ('Arnold', 0.0014062961588791488) |
| ('Paul', 0.0013903387391633749) |
| ('Z', -0.0013879262361200986) |
| ('R', 0.0013722776398363374) |
| ('Edwards', -0.0013666751184415124) |
| ('DEVITO-RODRIGUE', -0.0013575577856406141) |
| ('SHARELL', 0.0013245257548944818) |
| ('MAGRAS', -0.0013151021308263 29) |
| ('GRULLON', -0.0013114917255550751) |
| ('M', 0.0013008313889095752) |
| ('E', -0.0012960629778131011) |
| ('Herrera', 0.0012807023155936021) |
| ('Hatfield', -0.0012728655119864842) |
| ('JOSE', 0.0012579291472865938) |
| ('GRULLON', -0.0012442560742293395) |
| ('Guzman', -0.0012421304963356103) |
| ('ISLA', -0.0011877015530645629) |
| ('Donaldson', 0.0011866710733414714) |
| ('CAMPBELL', -0.0011763569827931183) |
| ('BONNER', 0.0011717232333539194) |
| ('BARNABY', 0.0011673858799001104) |
| ('Theresa', -0.0011670397955275932) |
| ('TAVERAS', 0.0011663768365565 38) |
| ('SANTORO', 0.0011471322594295156) |
| ('Kaufman', 0.0011353931875683026) |
| ('LIMATO', -0.0011303816355485903) |
| ('Derek', -0.0011286176368469525) |
| ('HARGRAVES', 0.0011070245601237428) |
| ('Cain', -0.0010901525656780155) |
| ('Allison', -0.0010767208109246239) |
| ('ANDERSON', -0.0010737612028428446) |
| ('B', -0.0010572042718499158) |
| ('Kathryn', 0.0010530100099604 39) |
| ('Matthew', -0.0010439297154172847) |
| ('J', -0.0010404575139248491) |

**Table 16: LIME Results - Products**

| 100 words |
|---|
| ('Protector', -2.6256731385071901e-06) |
| ('Ramps', -2.3891930611410793e-06) |
| ('Linen', -2.3436619972902771e-06) |
| ('90', -2.1822042687207914e-06) |
| ('Diced', -2.1095900325115041e-06) |
| ('All', -2.0906144044229118e-06) |
| ('60', -2.0875366521500831e-06) |
| ('Double-sided', -2.0660697678161833e-06) |
| ('C', -2.0519284562773661e-06) |
| ('Lemon', -1.9946896081845479e-06) |
| ('Oil', -1.9325735219645206e-06) |
| ('Close-up', -1.9271269704153383e-06) |
| ('Sugar', -1.9179601088354763e-06) |
| ('MELHORAL', -1.8979716223716453e-06) |
| ('Hair', -1.8921892830713871e-06) |
| ('120', -1.8880829607139037e-06) |
| ('Dining', 1.8877568586110512e-06) |
| ('LECHE', -1.8767255359194773e-06) |
| ('Definition', -1.869100759258901e-06) |
| ('Xylitol', -1.8610120962086523e-06) |
| ('Tomatoes', -1.8416359579577202e-06) |
| (""Tablets', -1.8281706632805249e-06) |
| ('Chewing', -1.8230891540283091e-06) |
| ('Lung', -1.8074822485009271e-06) |
| ('Detoxitech', 1.8062111028529713e-06) |
| ('Soft', -1.7804751654310493e-06) |
| ('Squeaksters', 1.7401255137497323e-06) |
| ('Butter', -1.739962185491471e-06) |
| ('Achari', -1.7387419284086943e-06) |
| ('Core', -1.7290335400193742e-06) |
| ('STYLE', -1.7044688391325595e-06) |
| ('Back', -1.7030634397438147e-06) |
| ('Predator', -1.6880776452748697e-06) |
| ('TB', -1.6848665630139052e-06) |
| ('Chelator', -1.6518442496193498e-06) |
| ('vcaps', 1.6465199109375219e-06) |
| ('L', -1.6387403257462667e-06) |
| ('200-RIZOS', -1.6090654576430883e-06) |
| ('Enema', -1.5974708730980886e-06) |
| ('Tablets', -1.5902977283374838e-06) |
| ('Just', 1.588204272712646e-06) |
| ('Elite', -1.5335638564684069e-06) |
| ('1x10', -1.5145173169796078e-06) |
| ('Height', -1.4925871373063442e-06) |
| ('Cooking', -1.4901305224565788e-06) |
| ('(Set', -1.4823013966042235e-06) |
| ('Tabs', -1.476077267303884e-06) |
| ('Amish', 1.4752362136062863e-06) |
| ('Pea', -1.4722668845971375e-06) |
| ('5', -1.4665139192096464e-06) |
| ('100', -1.4640129945103964e-06) |
| ('Peach', -1.4526914768189861e-06) |
| ('Lime', -1.450068051045915e-06) |
| ('Moxie', -1.4288992773748449e-06) |
| ('0.2-0.5%', -1.4191096791622887e-06) |
| ('Sunblock', -1.4177367457820354e-06) |
| ('CLARA', 1.4069510240662509e-06) |
| ('PHILIPS', 1.4042306274615911e-06) |
| ('Vegetarian', -1.3929175179679644e-06) |
| ('German', 1.3767813528298174e-06) |
| ('3', -1.3763715193717075e-06) |
| ('Exceptional', -1.3734769604219943e-06) |
| ('And', 1.3669455406460257e-06) |
| ('Margarita', -1.359951026740795e-06) |
| ('de', -1.3587517644007717e-06) |