Unsupervised Training Data Generation

Tejas Hasarali, Yevgeniy Ryasnianskiy, Romario Estrella-Ramos
Department of Computer Science
Florida State University

***Abstract*—This paper considers the problem of efficiently obtaining relevant data given a large collection of web tables. Specifically, we look into generating training data in an unsupervised fashion with the goal of obtaining useful and homogeneous groups of data. This paper presents a technique for the segmentation of data drawn from a large collection of unorganized web tables. The technique has three main steps: Data pre-processing utilizing NLP methods, word clustering using Word2Vec and document clustering with TF-IDF and DBSCAN algorithms.**

***Keywords*-Data Clustering; Unsupervised Training; Natural Language Processing (NLP) Lemmatization; Term Frequency Inverse Document Frequency (TF-IDF); Density-based spatial clustering of applications with noise (DBSCAN); Word2Vec; Autoencoders; JavaScript Object Notation (JSON)**

I. Problem Statement

Random data obtained in the real world, including from the web, very rarely has any relevant structure. Thus obtaining any useful training sets based on similarities in a large data set is difficult and time consuming. Given a very large collection of random web tables, how can one efficiently obtain useful sets of data. This work proposes a method for generating training data in an unsupervised way in order to obtain relevant clusters of data that can become of use. The method proposed utilizes NLP methods and data clustering algorithms to get use able training sets.

II. Solution

A. Architecture

1) Data Extraction: The data was extracted from the MongoDB database in JSON format, it was then split into 30 chunks of size 300,000.

2) Data Pre-Processing: The obtained data chunks was pre-processed in python using using Natural Language processing libraries.

- The relation entry of the JSON represents the web tables in JSON array format, which was flattened based on the type of table (Entity, Relation, Layout, Matrix or Other) to convert it into a document form.
- Then the unnecessary information like hasHeader, hasKeyColumn, headerPosition, headerRowIndex, keyColumnIndex, lastModified, pageTitle, recordEndOffset, recordOffset, s3Link, tableNum, tableOrientation, tableType, textAfterTable and textBeforeTable were removed to obtain the JSON with just id, relation, title and url.

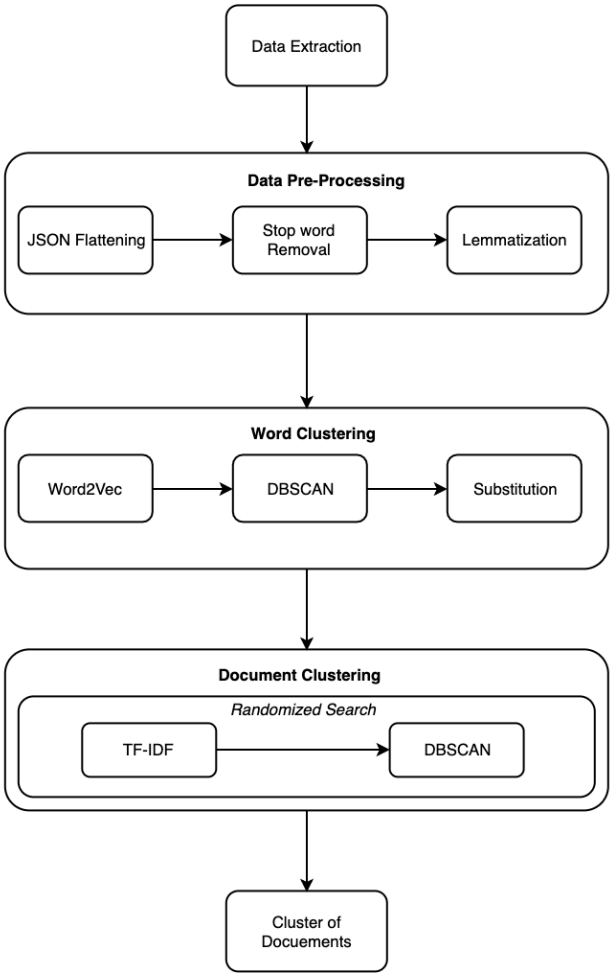

Figure 1. Arhietecture Diagram

- The relation attribute-value pair was processed using NLP to remove stop words, remove characters that are usually not used in English language and to perform lemmatization.
  - *Stop words:* They are nothing but useless words for clustering and search operations. These are commonly used words like a, an, the, is in English language. The stop word list was obtained from

Natural Language Toolkit and they were removed from the web table.
  - *Lemmatization:* It is a linguistic process of converting the words into their root words or dictionary form. The lemmatizer first finds the parts of speech of the word, then it will use the normalization rules based on the parts of speech to find the lemma/root word. This root word is then searched in the dictionary to check if the produced lemma is a valid word. Hence it will always produces a valid root word.
- The relation column of 30 data chunks of size 300,000 each, for a total of 9,000,000, documents from the pre-processed data was tokenized using Natural Language ToolKit to create document wise list of words, which was used to train a language model, specifically Word2Vec. This model is used to reduce the complexity of the dataset by grouping words and substituting a single word for all words in a group. We choose the most frequently occurring word in the group to maximize human readability after in subsequent steps. The language model ensures that this substitution is meaningful.
  - *Word2Vec:* Is a model for learning meaningful vector representations of words in a corpus of text [8]. Vector representations have the property that words with a similar meaning will be collocated. Meaningful transformations are also possible; however, we do not take advantage of this property.
  - *DBSCAN:* The DBSCAN algorithm, which is explained in the later part was applied on the word vectors obtained from Word2Vec language model to generate cluster of words.
  - *Substitution* The cluster obtained from the DBSCAN was used to simplify the data by substitution. In which, every word in the cluster is substituted by the most common word in that cluster. For instance, the words *cost*, *price*, and *value* (in order of frequency) would all be replaced by the word *cost*.
- The simplified word clustered web-tables are passed to TF-IDF for document clustering.

4) Document Clustering: The documents are clustered using using TF-IDF and DBSCAN algorithms. Here 1,000,000 documents are clustered after the application of the prepossessing steps. The 1,000,000 are a subset of the 9,000,000 documents that were used to generate the word vector model.

- *TF-IDF:* It stands for Term Frequency Inverse Document Frequency, which is to used find the importance of all the words in a document. The importance of the word increases with number of occurrences in the document. The TF-IDF can be calculated by multiplying two terms - Term Frequency and Inverse Document Frequency.
  - *Term Frequency (TF):* As the word suggests it estimates the number of occurrences of all the words in the document and it can be calculated by dividing number of occurrences of each word in the document with total number words in the document. It is divided by total number of words to accommodate documents of different lengths.
  - *Inverse Document Frequency (IDF):* It estimates the importance of all the words in the document. The words which occurs in all the documents have little meaning compared to the words which are unique to some documents. Hence to scale up the rare words we will calculate IDF by dividing the total number of documents with number of documents with a particular word.

  TF-IDF was used to convert the document into vectors to cluster similar documents using DBSCAN to generate data for supervised learning.
- *DBSCAN:* It stands for Density-based spatial clustering of applications with noise. It is a data clustering algorithm, which clusters the points that are close to each other while eliminating points that lie alone as outliers. The points which have a certain user defined number of neighboring points can be called as core point. The points which are alone without any neighbors can be called as outliers. The algorithm lets the user specify the distance between two points to be considered as neighboring points and the number neighboring points for a point to be called as core point. The algorithm starts by selecting an arbitrary point as the initial point, then it tries to find user defined number of neighboring points, if it finds sufficient number of points then it will start a cluster, otherwise the point will be marked as noise and selects a different point to find a cluster. It should be noted that the initial point could be included in a different cluster in the later stages even though it is marked as noise. The process continues until all the neighboring points are explored. Then it will select a different unexplored point to start a new cluster. The DBSCAN outputs the group to which each document belongs to

5) Parameter Optimization: DBSCAN, Word2Vec, and TFIDF all have hyperparameters which can significantly affect performance depending on the application. To optimize these parameters, we search through a parameter space and evaluate our clusters against a novel metric designed for unsupervised learning algorithms.

- *Random Search:* The algorithm is run for several parameter sets that are randomly sampled from a predefined probability distribution. This distribution is can be

seen in Table I and Table II-A5. For each parameter set the entire algorithm is run on the data. The resulting document clusters are evaluated against a metric that we have defined for this application. Typically, this process is applied to supervised learning algorithms where accuracy is evaluated against a test set. The word clustering was evaluated manually based on the percentage of words clustered by DBSCAN and the quality of the of the obtained clusters.

- *Score Metric:* For the groups obtained through data clustering an algorithm was designed to give each cluster a score that is based on their usefulness in our task. This score measures the average heterogeneity of the generated clusters. The scoring algorithm runs within the randomized search in order to score the clusters and compare the randomly sampled parameter sets.
  - *Algorithm:*
    Let S be the score given to a group of clusters, where the clusters are created from a subset of the data. Also let X represent the data passed into the algorithm, this includes noisy data not in any cluster
    Let,
    $$A = (\sum_{i \in X} i) + (\sum_{i \in X, i \neq -1} i) \tag{1}$$
    be the uniqueness of the URLs in each cluster, where i represents documents in the data X. The first sum i is the number of unique URLs in the given data set, while the second is the number unique URLs where i belongs to a cluster and is not noise.

    Let,
    $$B = \sum_{c \in C} \frac{i}{\mid c \mid} \tag{2}$$
    represent the similarity of data in each cluster, in other word a value representing how homogeneous the data is within a cluster. Note: i is the is total number of unique URLs in each cluster.

    We can now define the score algorithm
    $$S = \frac{A + \frac{B}{n} + n}{\mid X \mid} \text{ , if } \mid X \mid > 1 \tag{3}$$
    and
    $$S = 0 \text{ , if } \mid X \mid \leq 1 \tag{4}$$
    Note that the score is 0 for subsets of data of size less than or equal to 1. This constraint is added to indicate that not enough data was passed to the score algorithm. Also, n represents the total amount of documents that belong to a cluster (not noise).
    The score S for all the groups is a measure of how useful the clusters created are and is given to the random search algorithm to serve as a score for the parameters. If there are too many or too few clusters some of the groups many display much narrower scores than the rest.

Table I
WORD CLUSTERING PARAMETERS

Parameter	Distribution	Best Value
Word2Vec min-count	2 - 10	10
Word2Vec size	100 - 400	300
DBSCAN Epsilon	0.01 - 0.75	0.25
DBSCAN metric	euclidean, cosine	cosine

Table II
DOCUMENT CLUSTERING PARAMETERS

Parameter	Distribution	Best Value
DBSCAN Epsilon	0.2 - 0.9	0.6
DBSCAN Samples	2 - 24	8
TFIDF Features	10,000 - 800,000	180,000
TFIDF nGram	(1,1) - (2,3)	(1,3)

III. RESULTS

9,000,000 of the Pre-processed documents were used to train the word2vec model. The selected parameters can be seen in the table Table I. Here we have considered the words which occur more than 10 times and a window size of 5 was used. The window size helps in summarizing the word based on the information of 5 words which are present before and after it. The obtained word2vec model has 13,27,851 unique words with 300 size vectors representing each word.

From this, vectors of 721,701 unique alphanumeric words present in the selected 1,200,000 documents were used for word clustering using DBSCAN algorithm. The DBSCAN clustered 23.14% of the data, with cluster size ranging from 149,168 to 3 of which a sample of clusters can be seen in the table section III. The words in the cluster was replaced by the most common word in that cluster, which simplifies the data. It is then passed to TF-IDF and DBSCAN for document clustering.

801,303 documents of 1,000,000 were clustered. With cluster sizes ranging from 160,325 to 2 with a mean size of 482. The largest cluster is composed of web-tables that have dates, times, and math formulas. The next larges cluster has a size of 1994. In Table III we provide a sample of clusters that are generated. The average heterogeneity score of the generated clusters is 0.29. This indicates that around 7/10 entries in a cluster belong to an already represented website. This ratio is even greater for large clusters. Two of the example clusters were not unique, there exist other clusters for shoes and football. This is one limitation of

Table III
DBSCAN OUTPUT OF WORD CLUSTERING

Group Size	Max Word	Cluster of Words	Description
245	charizard	sneasel, zubatdodrio bulbasaur, staryu dugtrio, nidoking shellos, combusken	Names of Pokemon
32	cabernet	smillon, beerenauslese, syrah, pinotage, counoise sangiovese	Names of wines
88	pearland	brackettville duncanville, monahans 77447, schertz hoston	Places and zip codes in Texas
67	ale	weihenstephan, dunkler moortgat, lagerbier unblend, kellerbier	Brewers and beer brands
64	err15	err24, err15 err20, err40	Errors with error Number

Table IV
DBSCAN OUTPUT OF DOCUMENT CLUSTERING

Group Size	Sample Web-Table	Description
680	us eur japan 4.5 - 5 35 n / 5.5 - 6 36 23 6.5 - 7 37 23.5 - 24 7.5 - 8 38 24.5 - 25 8.5 - 9 39.5 25.5 - 26 9.5 - 10 40.5 26.5 - 40	Shoes sizes from websites such as Juicy Couture, heels.com, SHOPBOP, Just Fabulous, Peltz Shoes, Sports Authority, Tobi, MLB.com, etc.
566	total % 4) evangelical presbyterian church (epc) 1 0.0 5) just presbyterian 1 0.0 7) presbyterian church america (pca) 13 0.4 8) presbyterian church usa (pcusa) 31 0.9 9)	Municipal demographics, data, and facts from Capitol Hill Neighborhood, Auraria Library, Denver Municipal Facts, City and Householder Directories, Paulding County USGenWeb etc.
828	total net yards 297 248 total plays 60 72 average gain 5.0 3.4 net yards rush 152 29 rush 33 19 avg . per rush 4.6 1.5 net yard pass 145 219 comp .	Football data from D3 football, What If Sports, Pro Football Reference, CBSSports, ESPN, UCLABruins, The Augusta Chronicle, ScoresandOdds etc.

this technique. Small variations in cluster syntax can result divergent cluster generation.

IV. RELATED WORK

Other approaches to clustering documents use optimization techniques to derive clusters. Abualigah et al. proposes using the Krill Herd algorithm to generate clusters [9]. The algorithm is biologically inspired by herding krill which attempt to minimize the distance between themselves and their food. Here individual agents are modeled and documents are substituted for food. Krill forage, are attracted or repulsed by other krill, and may move randomly. Krill evolve using genetic algorithms and individuals have different behavior. In this paper optimization algorithms are compared to clustering algorithms on several standard data sets. The data shows that the krill herd algorithm is superior; however, the authors do not mention the ability of this method to scale to larger data sets.

The authors of the paper [1] were seeking to find technical terms in texts, specifically patents, without having to rely on existing resources. The solution proposed for tackling this task involved the use of Natural language processing (NLP). As stated in the paper, a large part of our technological knowledge is encoded in patents, thus methods for finding and inferring information in them is of great importance. A novel method is proposed for labeling large amounts of training data in an unsupervised fashion. This method has three parts: training set generation, parameter selection and training of a specialized classifier and identification of terminology in documents. A method for generating high-quality training data in an unsupervised fashion is proposed in the research, which does not need pre-compiled resources.

The paper [2] seeks to show a method for combining subspace clustering methods and artificial neural networks. An unsupervised clustering system with improved performance is achieved from combining the two. This research utilizes existing subspace clustering techniques to generate unsupervised training set from data that comes from a union of sub-spaces. The set produced is then utilized to obtain a Convolution Neural Network model, which improves clustering accuracy of conventional subspace clustering methods. Generating the training data in an unsupervised way was completed by using sparse subspace clustering. This method can deal directly with data nuisances, such as noise, sparse outlying entries, and missing entries.

The Paper [3] considers the problem of subspace clustering under noise. The behavior of Sparse Subspace Clustering (SSC) is analyzed when presented with noisy data. SSC creates groups of objects that are related based on observation of their attributes. This means it uncovers the underlying structure of the data in order to cluster them. The authors provide a modified version of this method that is effective in correctly identifying the underlying subspaces, even with noisy data.

The paper [4] tries to improve the existing auto-encoder approach for data clustering, auto encoders are widely used in data clustering by compressing the complex high dimensional data into simple low dimensional data. The clustering can be easily performed on the low dimensional data as they are more separable compared to the high dimensional complex representation. But it suffers from major drawback of converting different kinds of complex data into the same low dimensional representation. The paper proposes a novel approach of creating K auto-encoders with different low dimensional representations. The data is passed through all the K auto-encoders to produce K different low dimensional representations, which is then passed through a simple neural network to select the best low dimensional representation for the given data. Using the selected low dimensional representation data is classified.

The paper [5] tries to cluster the humongous amount of news articles online to make, tracking of news development easier. The classic methods for text mining with the advancement of Natural Language Processing is using weighted frequency of words or Latent Dirichlet Allocation. But these methods fail to consider the meaningful association chain present in natural language and the hierarchical relationship between different news articles. With the introduction of powerful Word2Vec concept of representing words in the form of vectors, which keeps the innate natural language information, the text mining can be improvised. The paper represents the news articles using the Word2Vec representation, which is trained on the Wikipedia articles. Then the multi-scale community detection method is applied on the obtained Word2Vec representation, the result is used to find the similarity graph of document vectors. The proposed method takes advantage of powerful Word2Vec representation and also groups the articles on different resolution levels.

V. Conclusion

Data obtained in the real world rarely has any relevant structure as the means to obtain large data sets is time consuming. Checking that the data retrieved holds a useful structure would take too much human resources. Thus, creating a technique to take raw unclassified data and separating it into training sets of homogeneous data is of great significance. Given a large collection of web tables how could relevant sets of data be extracted. This paper presented a technique for generating training data in an unsupervised fashion.

The method introduced makes use of several algorithm, including Word2Vec and DBSCAN, with the goal of producing training sets. These sets are the result of data preprocessing, word clustering, and document clustering. The resulting clusters, composed of documents, is the training data that can be of use.

VI. Future Work

For the training sets produced it would be of great value to automate the process giving each one a title or a brief description in order to identify its contents. This would allow the training sets to be identified faster for use in other applications such for training models for machine learning. There are several other clustering algorithms that could have been used in the approach presented. It would be of great benefit to test them in order to test and possibly improve the efficiency of our technique. Detecting and merging duplicate clusters is another technique that can be used to improve the quality of the clusters that are generated.

References

[1] Alex Judea, Hinrich Schütze, and Sören Brügmann. 2014. Unsupervised Training Set Generation for Automatic Acquisition of Technical Terminology in Patents. in *the 25th International Conference on Computational Linguistics: Technical Papers,* pg 290-300, Dublin, Ireland, August 23-29 2014

[2] Ali Sekmen, Ahmet Bugra Koku, Mustafa Parlaktuna, Ayad Abdul-Malek, Nagendrababu Vanamala. 2017. Unsupervised deep learning for subspace clustering. in *IEEE International Conference on Big Data (Big Data)*. 11-14 Dec. 2017

[3] Yu-Xiang Wang, Huan Xu. 2016. Noisy Sparse Subspace Clustering. in *Journal of Machine Learning Research 17*. pg. 1-41. 2016

[4] Dejiao Zhang, Yifan Sun, Brian Eriksson and Laura Balzano. Deep Unsupervised Clustering Using Mixture of Autoencoders. In *arXiv:1712.07788 [cs.LG]* 2017

[5] M. Tarik Altuncu, Sophia N. Yaliraki and Mauricio Barahona. Content-driven, unsupervised clustering of news articles through multiscale graph partitioning In *arXiv:1808.01175 [cs.CL]* 2018

[6] Gentile, Anna Lisa, Petar Ristoski, Steffen Eckel, Dominique Ritze, Heiko Paulheim. Entity Matching on Web Tables: a Table Embeddings approach for Blocking. In *EDBT*, pp. 510-513. 2017.

[7] Ristoski, Petar, and Heiko Paulheim. Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Web semantics: science, services and agents on the World Wide Web 36* (2016): 1-22.

[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *In Advances in neural information processing systems* (pp. 3111-3119).

[9] Abualigah, L. M., Khader, A. T., Hanandeh, E. S., & Gandomi, A. H. (2017). A novel hybridization strategy for krill herd algorithm applied to clustering techniques. Applied Soft Computing, 60, 423-435.