Can Language Help in the Characterization of User Behavior? Feature Engineering Experiments with Word2Vec

Eduardo Lopez
Information Systems
McMaster University
Hamilton, ON, Canada
lopeze1@mcmaster.ca

Kamran Sartipi
Department of Computer Science
East Carolina University
Greenville, NC, USA
sartipik16@ecu.edu

Abstract

[1] *Among the many significant advances in the area of deep learning, the Natural Language Processing (NLP) space holds a special place. The availability of very large datasets along with the existence of powerful computing environments have created a fascinating environment for researchers. One of the algorithms recently developed is Word2Vec, which enables the creation of embeddings (low-dimensional, meaningful representations of language that can be used for machine learning tasks such as prediction or classification). In this study, we experiment with Word2Vec and apply it to a different domain, i.e., representation of user behavior in information systems. We demonstrate how feature engineering tasks for user behavior characterization can be enriched by the use of NLP concepts.*

1 Introduction

Information Technology (IT) has enabled dramatic transformations in the way organizations execute their processes. Using information systems for the delivery of value is a competitive necessity across all industries, producing a trove of digital data that can be used for myriad purposes. Behavior is no longer an abstraction constrained to the physical world, but can also refer to the way in which people use the information systems at their disposal. The vast majority of users' interaction with information systems is captured in electronic documents – known as logs. These are usually system-specific, very large files that store actions, events and/or contextual parameters in the form of unstructured data. An analysis of these files may provide remarkable insights into the use of information systems.

This study experiments with a real-life, anonymized set of logs that capture the behavior of many users over a period of continued monitoring spanning 58 days. The objective is to extract from this data the key elements that would allow characterization of user behavior in information systems, and that can be used downstream in tasks such as prediction or classification.

This paper is organized as follows. Section 2 explores the different foundational concepts that support the analysis in this study. In Section 3 we describe the approach, delving into technology architecture, data structures and techniques. We describe our implementation in Section 4 and conclude our discussion with a summary of our contributions in Section 5.

2 Background on Machine Learning

In this section, we articulate some of the concepts that support the experiments described in this study. Under the broader umbrella of Artificial Intelligence (AI), machine learning – and more specifically deep learning – is demonstrating great success with many real-world applications [4]. A large number of achievements in areas such as computer vision or language have come close or surpassed human performance as measured by standard tests [5].

Perhaps one of the most remarkable developments in the Natural Language Processing (NLP) space is the Word2Vec algorithm. It was created by a team of researchers led by Tomas Mikolov in 2013 [3]. Word2Vec is best explained with an example. The following words (alphabetically indexed) exist in a 5,000-word vocabulary: Man (2203), Woman (4390), King (1348) and Queen (3100). Each word may be represented by a one-hot encoded sparse vector, where only the index position of that word has a value of 1. Figure 1 depicts this scenario.

Although these features are numeric, and suitable to feed mathematical models, the sparse representation does not enable comparison between two words as the similarity metrics are meaningless in this context. In contrast, Word2Vec

[1] DOI reference number: 10.18293/SEKE2020-117.

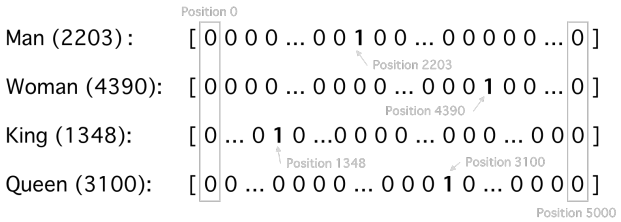

Figure 1. Sparse representation using one-hot encoding for a 5,000 words vocabulary.

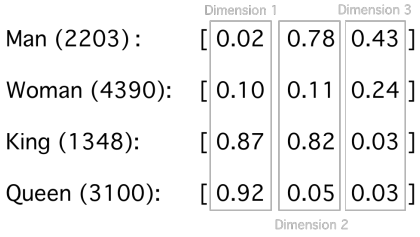

Figure 2. Word2Vec 3-dimensional representation of the words.

ingests an existing document (that uses the 5,000-word vocabulary) and produces a dense representation of the words in a lower-dimensional space. The Word2Vec does it by training a shallow neural network with two layers to predict a word given the words (i.e. context) around it in the inputted document. Once the training period is completed, the output layer is discarded, and the final weights are returned as the new representation. For example, a Word2Vec configured to yield three dimensions may produce a vector space that enables comparison between vectors, as per Figure 2.

3 Approach

Using the Word2Vec concepts in a suitable database is a remarkable opportunity that this study pursues. As was explained previously, the intent is to characterize users' behaviors using the tools and techniques from the NLP space. A very large and completed dataset is available from the Los Alamos national laboratory in the United States [2]. It contains the logs from multiple devices running on the same network over a period of 58 days. It includes authentication actions, Domain Name Service (DNS) calls, routing flows and programs started or ended by users. Our attention in this experiment revolves around the programs (i.e. processes) log. It has more than 426 million records uniquely identifying what programs were used by the users.

Manipulating this dataset requires computing power and

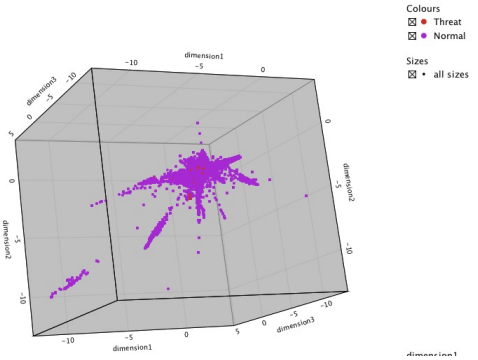

Figure 3. 3-dimensional feature space produced from the raw data through the Word2Vec algorithm.

software beyond the commonly offered in end user workstations. We execute the processes described in this study in a Linux cluster running Apache Spark [1], a unified data science open source tool that implements many of the best-known AI algorithms including Word2Vec.

4 Experimentation

The following experiments are performed using the approach described.

4.1 Feature engineering with Word2Vec

A single program (i.e. process) is coded with the letter 'P' and an integer number. We define a process profile $\vec{PP}$ as a set of processes that are executed by the user in any given hour. The relationship between a process and a process profile is similar to that of a word to a document. Thus, we proceed to use the process profiles as the document to be processed in Word2Vec. There are a total of 2,097,198 records capturing the process profiles ('documents'). We use the Word2Vec implementation in Apache Spark which averages each document when finding the lower-dimensional representation of each word (i.e. process in the case of this study). We experiment with several different dimensional spaces: 1, 2, 3 dimensions (which can be plotted) as well as 10, 100 and 1000.

The 3D vector feature space is depicted in Figure 3. The two classes ('threat' and 'normal') are represented by the red and blue data points.

As the intent of this experiment is to assert whether user behaviors can be extracted using Word2Vec, we proceed to cluster the data in order to best understand if there are regularities that can be detected. The feature set produced by Word2Vec is clustered using the K-means technique. Using

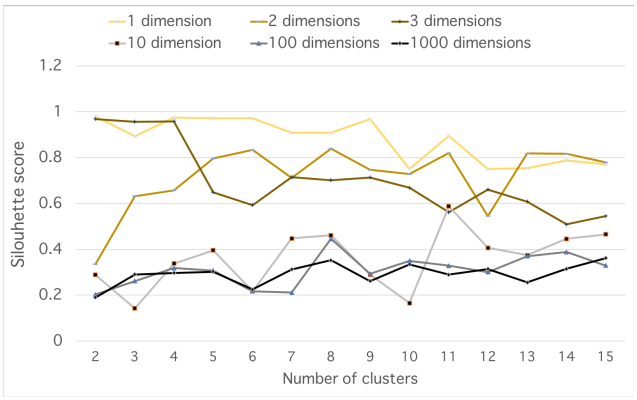

Figure 4. Cluster quality for 1-, 2-, 3-, 10-, 100- and 1000-dimensions, Word2Vec-created vector spaces

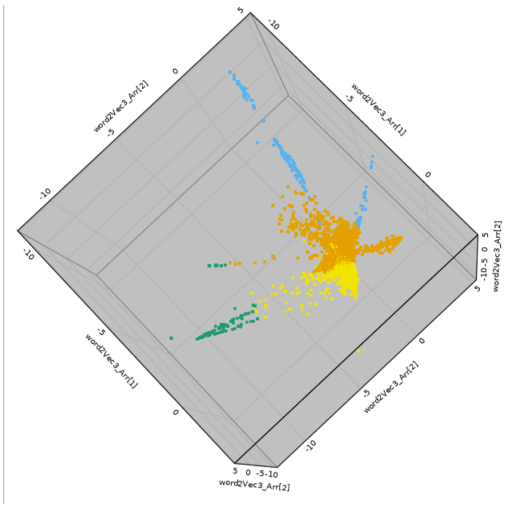

Figure 5. K-means clustering for three-dimensional Word2Vec vector space

K-means clustering we establish the silhouette scores from 2 to 15 clusters for each of the Word2Vec vector spaces calculated: 2-, 3-, 10-, 100- and 1000-dimensions. Figure 4 displays the different scores.

There are multiple remarkable elements in this depiction. First: the larger dimensional spaces (10,100 and 1000) found with Word2Vec do not appear to cluster the data well, although they begin to improve as the number of clusters grow (which is to be expected as very high clustering overfits the data). Second: the one dimensional Word2Vec clusters the data well (which is to be expected given the simplicity of clustering scalar numbers). However, using only one dimension neglects the complexity behind user behavior and – in trying to measure it – reduces it to one number only. Using 2 or 3 dimensions enable a better, more textured interpretation and still produce high quality clusters. A third critical point is – knowing that K-means produces different clusters when ran repeatedly as it departs from different centroids – the clustering is performed multiple times, and the results are averaged for generalization purposes.

We select the 3-dimensional vector space, and the maximum number of clusters that yield a 0.95+ score, which is 4 clusters. Three dimensions can represent a wide range of user behaviors, it is easily plot-able and permit the dividing of process profiles into four distinct and well-delineated groups. This can be observed in Figure 5, where the different clusters are depicted with four different colors. We conclude that the features engineered from the data are suitable for clustering activities, reflecting the utility of the feature space estimated.

4.2 Machine learning: classification

The second activity that is performed to assess the utility and effectiveness of the features engineered is classification. The dataset includes labeled data identifying whether the activity was performed by a regular user or by a user belonging to the ”red team”, i.e., users behaving abnormally. This is a supervised learning model, in which the objective is to verify that the features extracted are suitable for identifying normal vs. abnormal behavior in the feature set.

Every user in the feature set has distinct user behaviors that the classification exercise analyzes. Thus, the logistic regression model needs to be run for each user and not for the total 2.7M records. This means that more than 11,000 logistic regressions (i.e. classifiers) are instantiated and calculated with the labeled data. The feature dataset is partitioned in a training set (80% or approximately 2M records) and a test set (20% or approximately 500K records).

Figure 6 depicts the classified data along the three dimensions calculated with Word2Vec. The color and size convey the probability (blue=1) of an observation being a threat.

It is now possible to calculate how good the classifier was in assigning the correct labels. The following confusion matrix captures the results when each observation is labeled a threat for probabilities higher than 0.5.

The total number of records classified (the test feature set) is 537,280. The ground truth (i.e. known labels) have 161 records labeled as ’threat’. The 3D-Word2Vec logistic classifier predicted 140 records as threats, 87 correctly and 53 incorrectly. Given the rarity of records labeled as threats, the overall accuracy of the classifier is not a good indicator

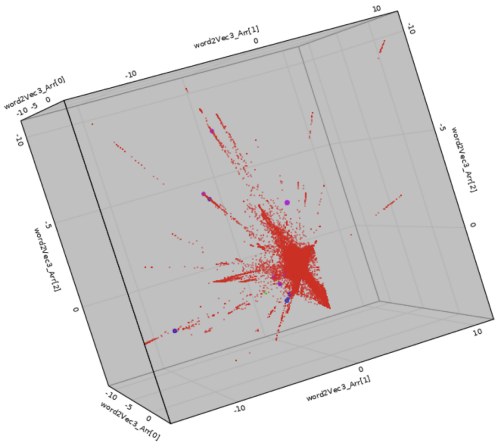

Figure 6. Classified (logistic regression) Word2Vec 3D vector space Color=probability of observation being a threat

Row ID	Threat	Normal
Threat	87	74
Normal	53	537066

Figure 7. Confusion matrix (threshold = 0.5) for the logistic classifier.

on its prediction quality. The sensitivity (also called true positive rate) and specificity are calculated as

$$sensitivity = \frac{correct\ threat\ predictions}{total\ number\ threats} = \frac{87}{140}$$

$$specificity = \frac{correct\ normal\ predictions}{total\ number\ normals} = \frac{537,066}{537,140}$$

The classifier has virtually perfect specificity (99.98%), with an adequate 62% sensitivity. It is important to note that the probability of randomly picking an observation labeled as a threat is $\frac{140}{537,280}$ or 0.0002%. It is, therefore, possible to conclude that the classifier built with the features engineered are a very good source of information to identify the threat labeled records.

5 Conclusion

In this study we explore the feature engineering aspects – extraction, transformation and selection – of variables that contain sufficient information for downstream analysis processes such as clustering and classification. We can conclude that using a multidimensional representation of the programs enable suitable characterization of user behavior. The use of process profiles (i.e. processes started or ended in a given hour by a user) can be equated to language documents that contain words. Extending the analogy to the Word2Vec algorithm allows for the transformation of feature vectors into a dense representation.

Given the results of the different K-means clustering activities, the silhouette scores indicate that three dimensions suffice for the grouping of user behaviors, even enabling plotting for added understanding of the dynamics.

References

[1] Apache Spark™ - Unified Analytics Engine for Big Data. https://spark.apache.org/.

[2] Los Alamos National Lab: National Security Science. https://www.lanl.gov/.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[4] Raymond Perrault, Yoav Shoham, Erik Brynjolfsson, Jack Clark, and John Etchemendy. Artificial Intelligence index - 2019 annual report.

[5] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *arXiv:1905.00537 [cs]*, Feb. 2020.