K. David Pearce
HUDM 5026

Final Project: Natural Language Processing with R

Introduction

One of the most challenging aspects of interdisciplinary science is associating words and concepts from one academic discipline to another. For example, if a neuroscientist wants to understand a particular cellular interaction in terms of the latest advancements in physics, but has minimal training in physics, then the neuroscientist might encounter significant barriers relating journal articles from physics to brain research.

One method of quickly breaking down these barriers might be with the assistance of machine learning and natural language processing (NLP) algorithms. In general, machine learning is a type of statistical analysis where an algorithm is trained with data rather than being explicitly programmed. More specifically, a system is presented with many examples relevant to a task, which allows it to find statistical structure within the examples, and then come up with rules for automating the task (Chollet, 2018).

Machine learning techniques fall into one of two categories: supervised learning or unsupervised learning. Supervised methods require large, hand-labelled datasets for training, and are often associated with prediction problems ("fancy regression"). This is akin to giving a student a set of problems along with their solutions and telling that student to predict how other problems might be solved. Unsupervised methods do not require human labelling or supervision, and are often applied to clustering problems ("fancy clustering"). This is like giving a student a set of patterns and asking that student to figure out the associations between those patterns (Louridas & Ebert, 2016).

For example, Shuffrey (2019) used an unsupervised learning package in R Studio called clValid, which is a type of hierarchical clustering, to identify patterns of maternal prenatal alcohol exposure. Specifically, Shuffrey's analysis identified nine clusters of mothers within her sample who used alcohol during pregnancy, exposing an unexpected cluster of mothers who had not consumed alcohol during the first trimester, but began doing so during the second trimester.

Recently, Tshitoyan et al. (2019) showed that academic journal articles can be efficiently encoded as word embeddings (vector representations of words) using an unsupervised machine learning algorithm. Moreover, when applied to journal articles of a specific academic discipline – Materials Science in this case – the embeddings were able to capture complex chemistry concepts, such as the underlying structure of the Periodic Table, without any explicit insertion of chemical knowledge. The authors used an unsupervised machine learning algorithm called Global Vectors for Word Representation (GloVe), and its associated Python packaged called Word2Vec (Mikolov et al., 2013), for extracting knowledge and relationships from scientific literature. They demonstrated that an unsupervised method can recommend materials for functional applications several years before their discovery.

Selivanov (2016) created a package in R Studio called Text2Vec that functions similarly to Python's Word2Vec. When Selivanov applied his Text2Vec package to Geography articles from Wikipedia, the algorithm determined, for example, that the word vector for "paris" was most closely associated with the vector for "france." The algorithm does not know that Paris is the capital city of France, but it has determined that "paris" co-occurs so frequently with "france" within the data set that a relationship must exist between the two words. Interestingly, vector operations capture many linguistic regularities. For example, the GloVe algorithm can take a query such as "give me a word like king, like woman, but unlike man" and answer with "queen."

Following this logic, I wanted to know which words from neuroscience literature, if any, were associated with words from quantum mechanics. There is a fringe group of neuroscientists who believe that consciousness might be the result of neuronal activity occurring at the quantum level. When we apply Text2Vec to a corpus of Neuroscience and Quantum Mechanics journal articles, and then query the word "spin" (nuclear spin is a centerpiece of quantum mechanics) we find that the "spin" vector is associated with the vector for "hippocampal," which is the brain structure thought to be associated with memories. Interestingly, when we combine the vectors for "spin" and "hippocampal" the resulting vector is associated with "memory" and "phosphorus." Do these word associations truly represent significant relationships between Neuroscience and Quantum Mechanics worth investigating further, or are these word associations just nonsensical data produced from an insufficient data set? Maybe this is how a scientist begins to explore the associations between two disparate academic disciplines.

Data Description

The data set was created by combining five carefully selected academic articles, three from Neuroscience and two from Quantum Mechanics, into a single text file with 3,076 words. A data set with more academic articles might produce better quality results, although Tshitoyan et al. (2019) showed that the quality and domain-specificity of the corpus determined the utility of the algorithm. The text file was then read into R Studio as a character with one variable and 3,076 observations. Next, each observation was transformed into a token, which is essentially a word without any punctuation or spaces, and appended to a list consisting of 3,076 tokens.

After creating the list of tokens, the list was pruned so that it included only tokens that appeared in the corpus at least five times, since we cannot calculate a meaningful word vector for a word which occurred only once. This resulted in a list of 1,631 tokens. Finally, we applied R Studio's Text2Vec library to the list in order to construct a term-co-occurrence matrix (TCM) that was then factorized via the GloVe algorithm to produce a matrix of word vectors consisting of 1,631 x 50 dimensions, where one vector equals one distinct token (the variable) and its 50 most closely related vectors (observations).

Methods

The GloVe algorithm is essentially a log-bilinear model with a weighted least-squares objective. The premise of the model is that ratios of word co-occurrence probabilities have the potential for encoding some form of meaning (Pennington et al., 2014). GloVe consists of three steps, all of which are accomplished using one function, *fit_transform*, from the Text2Vec library.

Step one is generating a word co-ocurrence matrix *X*, where X_{ij} represents how often word *i* appears in the context of word *j*. Using the Text2Vec library, the *fit_transform* function scans the corpus by looking at each term and identifying the five context terms before and after the target term. The algorithm gives less weight for more distant words, using the formula:

$$decay = 1/offset$$

In step two, the *fit_transform* function applies soft constraints for each word pair:

$$w_i^T w_j + b_i + b_j = log(X_{ij})$$

Where w_i is the vector for the target word, w_j is the vector for the context word, and b_i , b_j are scalar biases for the main and context words, respectively.

In step three, *fit_transform* applies a cost function:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$$

Where *f* is a weighting function that limits learning to only common word pairs:

$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{x_{max}})^{\alpha} & \text{if } X_{ij} < XMAX \\ 1 & \text{otherwise} \end{cases}$$

The *fit_transform* function produces two vectors, a main and a context, which are nearly identical since the model is symmetric. Pennington et al. (2014) recommend combining the main vector with the context vector for more accurate results. Finally, we use the *sim2* (cosine similarity) function, which measures the cosine of the angle between two vectors projected in a multi-dimensional space, to tell us how similar our desired word vector is to all of the other word vectors.

Results

A query for the "spin" vector produces 50 results. The first neuroscientific word vector associated with "spin" is "hippocampal," with a cosign similarity of 0.31. For comparison, the word vector for "nuclear," often occurring in conjunction with "spin" as in "nuclear spin," has a cosign similarity of 0.63.

When the "hippocampal" vector is added to the "spin" vector, the results include "memory" with a 0.43 cosign similarity, and "phosphorus" also with a 0.43 cosign similarity.

Cosine similarity to 'spin'

spin	1.0000000
nuclear	0.6299533
Kadohisa	0.4458646
states	0.4385707
primate	0.4187601
induced	0.4118769
mouth	0.4063768
phosphorus	0.3956265
indirect	0.3865734
...	
hippocampal	0.3103817

Cosine similarity to 'spin' + 'hippocampal'

spin	0.8354289
hippocampal	0.7817568
nuclear	0.5648082
system	0.5116427
memory	0.4315895
phosphorus	0.4302941
primate	0.4123830
vacuum	0.3987979
spins	0.3924143
episodic	0.3910336

Conclusion

The Text2Vec library in R Studio provides a relatively easy and quick way for users to uncover word associations from a large corpus. A neuroscientist with minimal training in physics might find academic articles involving quantum mechanics to be difficult, if not impossible, to understand. With Text2Vec, however, the neuroscientist might discover previously unexplored associations, such as memory and phosphorus, without needing to fully understand quantum mechanics right away.

This study was limited by the number of academic articles included in the data set. Additionally, word associations are only a starting point for further investigation.

References

Chollet, F. (2018). Deep learning with python. Shelter Island, NY: Manning Publications Co.

Louridas, P., & Ebert, C. (2016). Machine learning. *IEEE Software, 33(5)*, 110-115.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *In Proceedings of Workshop at ICLR, 2013*. https://code.google.com/archive/p/word2vec/

Pennington, J., Socher, R., & Manning, C. *GloVe: Global Vectors for Word Representation*. https://nlp.stanford.edu/pubs/glove.pdf

Selivanov, D. (2016). *GloVe Word Embeddings*. http://text2vec.org/glove.html

Shuffrey, L. (2019). Identifying risk factors and perinatal markers for neurodevelopmental disorders. Talk presented at Teachers College, Columbia University, March 4th 2019.

Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K., Ceder, G., & Jain, A. (2019). Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature, 571*, 95-106. https://doi.org/10.1038/s41586-019-1335-8

Data Set

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, Samuel J. Gershman (2017). Building machines that learn and think like people. Behavioral and Brain Sciences, 40.

C.P. Weingarten, P.M. Doraiswamy and M.P.A. Fisher (2016). A new spin on neural processing: Quantum cognition, , Frontiers in Human Neuroscience 10, 541.

M.W. Swift, M.P.A. Fisher and C.G. Van de Walle (2018). Posner Molecules: From atomic structure to nuclear spins", Journal of Phys Chem Chem Phys, 20, 12373-12380.

Okihide Hikosaka; Susan R. Sesack; Lucas Lecourtier; and Paul D. Shepard (2008). Habenula: Crossroad between the Basal Ganglia and the Limbic System. Journal of Neuroscience 28(46), 11825-11829.

Rolls, E. (2015). Limbic systems for emotion and for memory, but no single limbic system. Cortex 62, 119-157.

Appendix

```
## Read the txt file into a Character with 3076 observations
```

```{r}
quantNeuroTxt <- read.delim("NeuroAbstractsPlusArticles.txt", header = FALSE)
```

```
## Load the text2vec library
```

```{r}
#install.packages("text2vec")
library(text2vec)
```

```
Registered S3 method overwritten by 'data.table':
  method           from
  print.data.table
```

```
## Create Iterator over Tokens
```

```{r}
tokens <- word_tokenizer(quantNeuroTxt)

```

```
argument is not an atomic vector; coercing
```

```
## Create vocabulary. Terms will be unigrams (simple words)
```

```{r}
it = itoken(tokens, progressbar = FALSE)
vocab <- create_vocabulary(it)
vocab <- prune_vocabulary(vocab, term_count_min = 5L)
```

```
## Vectorize the Vocabulary; Drop words mentioned < 5 times
```

```{r}
vectorizer <- vocab_vectorizer(vocab)
tcm <- create_tcm(it, vectorizer, skip_grams_window = 5L)
```

```
## Create Glove
```

```{r}
glove = GlobalVectors$new(rank = 50, x_max = 10)
```

Run the Analysis

```{r}
wv_main = glove$fit_transform(tcm, n_iter = 5, convergence_tol = 0,01,
n_threads = 8)
```

```
 INFO  [12:08:47.951] epoch 1, loss 0.1977
 INFO  [12:08:48.125] epoch 2, loss 0.1134
 INFO  [12:08:48.274] epoch 3, loss 0.0888
 INFO  [12:08:48.412] epoch 4, loss 0.0748
 INFO  [12:08:48.552] epoch 5, loss 0.0650
```

Analysis produces two vectors, a main and a context

```{r}
wv_context = glove$components
```

Combine the main vector with a transformation of the context

```{r}
word_vectors = wv_main + t(wv_context)
```

What word vectors are closest to "spin"?

```{r}
query1 = word_vectors["spin", , drop = FALSE]
cos_sim = sim2(x = word_vectors, y = query1, method = "cosine", norm =
"none")
head(sort(cos_sim[,1], decreasing = TRUE), 50)
```

What word vectors are closest to "spin" + "hippocampal"?

```{r}
query2 = word_vectors["spin", , drop = FALSE] +
    word_vectors["hippocampal", , drop = FALSE]
cos_sim = sim2(x = word_vectors, y = query2, method = "cosine", norm =
"l2")
head(sort(cos_sim[,1], decreasing = TRUE), 50)
```