Clustering NBA News:
Unsupervised Learning Methods

Jason Lee

Northwestern University, SPS
Natural Language Processing
2020SP MSDS 453-56

Abstract

Information is power in the sports betting industry. With the amount of news articles and information hitting the internet daily, professional sports bettors need to organize and prioritize the incoming information in order to react with speed before market prices adjust. As web scrapers scour the internet collecting NBA news articles, a document clustering algorithm is used to automatically sort these articles. The goal of this study is to utilize unsupervised learning methods to better understand and effectively sort the NBA news article corpus. There will be three overarching strategies to help with this goal: 1) Understand the corpus by using the articles as objects 2) Understand the corpus by using the terms in the articles as objects 3) Understand the corpus by topic modeling.

Keywords: Natural Language Processing (NLP), Clustering, Unsupervised Learning, NBA, Sports Betting

1. Introduction

The livelihoods of professional sports bettors revolve around staying current on all relevant sporting news. They must constantly be watching for breaking news alerts, injury updates, scores, and game specific analysis in order to execute advantageous investments. As focused web crawlers scrape the internet capturing up-to-the-minute news on each team, there needs to be an automated process to organize the incoming information. This is exactly what unsupervised machine learning techniques, such as document clustering and topic modeling can do.

The document classification model created in the previous project required manual labeling of each document in order to properly train the classifier (Lee, 2020a). While the end result was effective, the manual labeling is an extremely tedious process absorbing a disproportionate quantity of valuable time that could be spent in more profitable areas. Unsupervised machine learning techniques provide an alternative route to solving the initial problem a sports bettor faces with overwhelming amounts of news articles.

Unsupervised learning algorithms do not need labels or user inputs to accomplish their goals. They are able to read the incoming news articles and effectively sort them based on underlying similarities within the articles. This automated sorting process, coupled with the focused web crawlers, will allow any sports bettor the ability to digest the right information in a timely manner.

Building off of the corpus created by the focused web crawlers in the initial natural language processing (NLP) project (Lee, 2020b) and the document vectors created in the previous project (Lee, 2020a), the purpose of this study is to utilize unsupervised learning methods to better understand and effectively sort the NBA news article corpus.

There will be three means used to better understanding the corpus.

1. Understand the corpus by using the articles as objects.
2. Understand the corpus by using the words in the articles as objects.
3. Understand the corpus by topic modeling.

There will also be nine specific activities, or tasks, carried out over the course of this study connected to these three different approaches.

1.1. Activities:

The first activity prepares the NBA news article corpus. Activities 2-4 focus on understanding the corpus using the documents as objects, while activities 5-8 use the terms in the corpus as objects. The final activity uses topic modeling to try to further understand the corpus holistically.

1. Activity 1: Upload the three vectorized document matrices of the NBA news article corpus for the Analyst Judgment, TF-IDF, and Doc2Vec Embedding vectorization methodologies and prepare them for the following unsupervised learning tasks.
2. Activity 2: Perform partitioned cluster analysis (K-means) with the **documents as objects**. Utilize objective methods for determining the number of clusters (K).

3. Activity 3: Perform multidimensional scaling by way of the t-distributed stochastic neighbor embedding (t-SNE) algorithm with the **documents as objects**. Visualize the multidimensional scaling solutions in a two-dimensional space with clusters from Activity 2 as colors.
4. Activity 4: Analyze and compare the results from the multidimensional scaling and clustering techniques for the **documents** across the three different vectorization methodologies. Determine which of the three approaches provides the most clear-cut (interpretable) results.
5. Activity 5: Perform multidimensional scaling by way of the t-distributed stochastic neighbor embedding (t-SNE) algorithm with the **terms as objects**. Visualize the multidimensional scaling solutions in a two-dimensional space with clusters as colors.
6. Activity 6: Perform hierarchical cluster analysis with the **terms as objects**. Visualize the clustering solution with a dendrogram.
7. Activity 7: Analyze and compare the results from the multidimensional scaling and clustering techniques in relation to the **terms** used in the corpus. Determine whether the Analyst Judgment or the TF-IDF document vectorization approach provides the most clear-cut (interpretable) results.
8. Activity 8: Construct an ontology, or semantic network, of **terms** used in the corpus.
9. Activity 9: Perform topic modeling by using the Latent Dirichlet Allocation (LDA) algorithm.

The completion of these activities will yield a comprehensive understanding of the NBA news article corpus and help create a finely tuned document clustering model.

A management problem addressed with this study is the high cost of time and resources needed to manually organize countless articles while searching for the right information that could provide an edge for a professional sports bettor. Another problem this project will solve is the speed to act on the new information before the markets have time to adjust.

A document clustering model will be able to save a professional sports bettor countless hours by eliminating the manual effort needed to organize and read the various news articles and decide if it is useful or not.

A.I. Sports is the financial sponsor for this study and the clustering and topic models built herein will be their property. These models will be implemented through their company to better server their professional sports betting clientele (Lee et al., 2018).

There are three desired outcomes from this study:

1. Create a calibrated document clustering model.
2. Determine whether using multidimensional scaling or clustering provides the most clear-cut results.
3. Generate a reproducible Python notebook to easily share with colleagues.

2. Literature Review

Learning that takes place in nature by humans and animals can predominantly be categorized as unsupervised learning. This learning comes about by naturally putting the pieces together from what is in view. Unsupervised learning algorithms are used to discover patterns within unlabeled datasets. Due to the unsupervised characteristic, the results may vastly differ from what a person might expect. Yann LeCun, Chief A.I. Scientist at Facebook and recipient of the Turing Award for his work in Deep Learning, stated that the key step to attaining true A.I. is by solving the unsupervised learning problem (Patel, 2019).

Common machine learning approaches involve predicting a dependent variable based on its explanatory variables; this is called supervised learning. These algorithms learn by being fed lots of data with the dependent variable present. Unsupervised learning approaches are fed data with no dependent variable. The algorithm tries to make sense of the data by grouping, or clustering, together items that have similar features.

Document clustering models are powerful natural language processing (NLP) unsupervised learning algorithms that enable automated sorting and filtering systems to function independently of human intervention (Albon, 2018). They are trained on a given corpus and are able to find similarities and dissimilarities between the individual documents (Patel, 2019). In production these document clustering models are able to ingest a new document and properly sort it into a category.

2.1. Clustering Algorithms: K-Means

The K-Means algorithm uses the nearest neighbor between the items and groups them together into (K) number of clusters. The data scientist assigns the number of clusters to split the data into prior to training and then the K-Means algorithm will output a single cluster label for each item (Patel, 2019). The objective function when training a K-Means clustering model is to minimize the sum of the variations within each cluster (Patel, 2019). The algorithm strives to create roughly equal variations between the clusters during the training process (Albon, 2018).

There are three important assumptions made about the data if the desired outcome is to be reached when using the K-Means algorithm:

1. The clusters are convexly shaped.
2. The features are all equally scaled.
3. The number of observations are balanced between each group.

The K-Means algorithm is computationally expensive to train. In order to speed up the training time, each item is randomly assigned a starting cluster, which is reassigned and updated during the training process until convergence is achieved (Patel, 2019). When repeating the K-Means algorithm on the same dataset, the output of each iteration may be slightly different because of the randomly assigned starting cluster and setting a random seed is necessary to recreate the same results (Patel, 2019).

2.1.1. Visualizing Clusters: t-SNE

T-Distributed Stochastic Neighbor Embedding (t-SNE) is a process used to transform high-dimensional datasets into two or three dimensions allowing the data to be easily visualized (Patel, 2019). T-SNE reduces the dimensionality in a nonlinear way that retains the meaning of the original data; this can be thought of as an encoder or an embedding (Lane et al., 2019).

Unlike some dimension reduction techniques, t-SNE's nonlinear transformations allow the local structure of the data to be retained, while simultaneously uncovering high-level global structures (Van Der Maaten and Hinton, 2008). The newly created data points that are closer together in the two or three dimensional space are more similar, while the data points that are farther apart are dissimilar.

A critique of the t-SNE algorithm is that there is no stable solution because of the nonconvex cost function (Patel, 2019). This means that each initialization of the t-SNE algorithm on the same dataset will produce different results. Setting a random seed is needed when using t-SNE for reproducibility purposes.

2.2. Clustering Algorithms: Hierarchical

There are two primary methods used when clustering in a hierarchical manner. The first methodology is a top-down approach using divisive clustering algorithms (Izenman, 2008). The second methodology is a bottom-up approach using agglomerative clustering algorithms (Izenman, 2008).

In the agglomerative hierarchical clustering algorithm, each item begins as its own cluster and then is merged with similar items until the number of clusters remaining reaches the predetermined number of clusters, or a single cluster if no set number of clusters was designated (Albon, 2018).

The divisive algorithm for hierarchical clustering starts with a single cluster containing every item in the dataset. A split in the data is then made creating two clusters. The divisive algorithm recursively splits the groups until each item is isolated in its own cluster (Izenman, 2008).

2.2.1. Visualizing Clusters: Dendrogram

Hierarchical clustering algorithm results can be viewed with a tree-based diagram called a dendrogram. The dendrogram allows for extremely easy interpretations of the complex underlying unsupervised learning algorithm. Dendrograms use colors to represent clusters and lines to portray the relationships between items. The heights of the lines connecting items is representative of the similarity between them. Smaller heights correspond to strong similarities and taller heights correspond to less similarities between the items.

Figure 1 is an example of how a hierarchical clustering model of varies U.S. cities is visualized with a dendrogram. There are four main clusters represented by the different colors. Beyond the obvious color coded clustering, the interpretation of this diagram is straight forward.

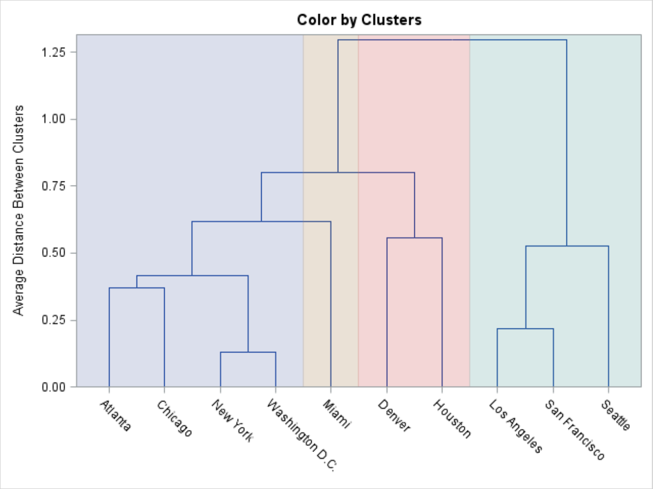

Figure 1: Example of a Dendrogram showing the hierarchical clustering of varies U.S. cities.

In the blue cluster on the left, New York and Washington D.C. are connected by a very short line meaning the cities have strong similarities within this model. Atlanta and Chicago are connected by a line that is just shorter than the line connecting them with New York and Washington D.C meaning they are slightly more similar than either city is with Washington D.C. or New York.

In the green cluster on the other side of the dendrogram, there is another short line connecting Los Angeles and San Francisco indicating strong similarities. The line connecting the green cluster containing west coast cities to the blue cluster containing east coast cities is the highest linkage in the diagram conveying the dissimilarity captured by the hierarchical clustering algorithm.

2.3. Topic Modeling: LDA Models

The Latent Dirichlet Allocation (LDA) model is a well established algorithm for topic modeling. This technique is able to determine which documents are most similar to each other by creating a simplified structure, or topic, within the raw text (Patel, 2019). These simplified structures, or topics, can be defined as collections of keywords that are representative and appear frequently within the given corpus (Prabhakaran, 2020). The topics found from LDA models help to quickly summarize and understand unstructured text documents (Patel, 2019).

3. Methodology

The methodology implemented during this study is sequential progressing through completing each of the nine activities mentioned in Section 1.1 of the introduction.

3.1. Document Corpus

The corpus used in this study was collected by focused web crawlers, or Spiders. The Spiders were released onto each National Basketball Association (NBA) team's official website moving from page to page collecting six important pieces of information from every news article they came across.

1. team = The name of the NBA team
2. url = The URL where the article is found
3. tags = The topic tags for the article
4. title = The title of the article
5. date = The date the article was posted
6. article = The complete news article

The topics contained in this corpus are wide ranging. There are articles written about the team's humanitarian efforts, potential trades, pre-game/match-up analysis, injury updates, post-game analysis, deep dive player specific topics, player written articles, front office management news, team perception/fan sentiment, fan outreach, and miscellaneous articles.

This natural language processing (NLP) project is a continuation from the previous NLP project utilizing the matrices created from the three document vectorization approaches (Lee, 2020a).

The three document vectorization approaches are as follows:

1. Analyst Judgment
2. TF-IDF
3. Neural Network Embedding

Only the document vector with 300 dimensions for each approach will be used throughout this study. There are 751 total NBA news articles in the corpus.

3.2. Dependent Variable

While unsupervised machine learning algorithms do not need a dependent variable, A.I. Sports' goal with these models is to effectively filter and sort the NBA news articles in a way that adds value to professional sports bettors. The document clustering models built during this study will be partially evaluated based on how well they are able to differentiate between relevant and irrelevant news articles. The dependent variable is a binary flag.

$$1 = Relevant$$

$$0 = Irrelevant$$

Topics that are included in the positive class include potential trades, pre-game/match-up analysis, injury updates, post-game analysis, and team perception/fan sentiment. Everything else will be contained in the negative, or Irrelevant, class.

Out of the 751 total news articles, 275 (36%) articles are classified as relevant to a sports bettor and 476 articles are classified as irrelevant.

4. Computational Experiment and Results

The entire Python code for this project will be attached to this paper, or can be reproduced by cloning the project's Google Colaboratory Notebook at this url:

Google Colab Link

The Python script begins by uploading the previously processed NBA news article corpus, followed by nine activities. Each activity is sectioned off in the notebook for ease of navigating. The Doc2Vec embedding vectorization method does not directly relate to specific terms the way that the Analyst Judgment and TF-IDF mothodologies do and will be disregarded in activities 5-8. The final activity does not use the document vectors and uses the raw text of each document in the corpus for topic modeling.

Activity 2 of the Python code is an adapted version of the cluster analysis example by Paul Huynh on the Reduced Reuters dataset (Huynh, 2020). Activity 9 of the Python code is an adapted version of a topic modeling tutorial by Selva Prabhakaran using the Gensim library (Prabhakaran, 2020).

4.1. Activity 1: Vectorization

Each document in the corpus was converted to a vector of numeric values. The previous project created vector lengths of 50, 150, and 300 for each of the three vectorization methodologies (Lee, 2020a). The vector length chosen for this study is 300 to provide the most depth and largest vocabulary for the clustering experiments that follow.

4.1.1. Analyst Judgment Vectorization

The Analyst Judgment methodology to vectorizing documents uses word statistics from the corpus. To easily generate the word statistics for this project, the CountVectorizer function from the SKLearn package in Python was used (SKLearn, 2020a). This function tokenizes the terms in the document and proceeds to count the number of times each token was used.

4.1.2. TF-IDF Vectorization

The Term Frequency-Inverse Document Frequency (TF-IDF) approach is similar to the analyst judgment, in that they both use word statistics from the corpus. For this study, the TfidfVectorizer function from the SKLearn package was used to calculate the TF-IDF values for each term in the corpus (SKLearn, 2020c).

4.1.3. Neural Network Embedding Vectorization

The final methodology to vectorizing the documents in the corpus is training a neural network embedding algorithm using Doc2Vec provided by way of the Gensim package in Python (Řehůřek and Sojka, 2010).

4.2. Activity 2: Clustering

The central goal in this activity is to understand the corpus by using the documents as objects. The vectors created in the previous activity were used as inputs into the clustering algorithms. After some experimenting, the number of clusters chosen for this and the next activity was six.

4.2.1. Analyst Judgment (K-Means)

	C-1	C-2	C-3	C-4	C-5	C-6
Relevant	117	118	0	5	0	35
Irrelevant	118	315	1	32	1	9
Total	235	433	1	37	1	44

Table 1: NBA News Articles per Cluster for the Analyst Judgment vectorization methodology.

The Analyst Judgment clusters were heavily skewed with the bulk of results residing in clusters 1 and 2. Cluster 3 and 5 each had only a single document classified.

Cluster 6 proved to be the best cluster for a sports bettor to read containing 79.5% relevant NBA news articles. Cluster 1 had 49.8% relevant articles, which is a slight improvement from the initial 36.6% distribution.

4.2.2. TF-IDF (K-Means)

	C-1	C-2	C-3	C-4	C-5	C-6
Relevant	16	66	120	15	32	26
Irrelevant	172	16	77	70	111	30
Total	188	82	197	85	143	56

Table 2: NBA News Articles per Cluster for the TF-IDF vectorization methodology.

The document counts of the TF-IDF clusters were more evenly distributed across the six clusters compared to the other document vectorization methodologies. Table 2 provides the details for each cluster in this model.

Cluster 1 was able to filter down the relevant articles to only 8.5% with a 188 sample size making it a valuable cluster to avoid for a sports bettor. Another cluster to avoid reading was cluster 4 containing only 17.6% of its articles as relevant. Cluster 2 was able to peak at 80.5% relevant news articles. Cluster 3 contained the absolute maximum for relevant news articles with 120 making up 60.9% of the total articles in the cluster. Cluster 5 had 22.4% relevant articles making it a decrease of 14.2% from the original distribution and finally cluster 6 only had an increase of roughly 10% coming in it at 46.4% relevant.

Four of the six clusters in this model were able to create a sizeable lift discriminating between the relevant and irrelevant NBA news articles over the baseline 36.6% relevant proportion.

4.2.3. Doc2Vec Embedding (K-Means)

	C-1	C-2	C-3	C-4	C-5	C-6
Relevant	4	153	0	0	11	107
Irrelevant	5	38	34	6	85	308
Total	9	191	34	6	96	415

Table 3: NBA News Articles per Cluster for the Doc2Vec Embedding vectorization methodology.

Similar to the Analyst Judgment approach, the six clusters created with the Doc2Vec embedding vectors were heavily skewed with cluster 6 containing over half of the documents in the entire corpus. Clusters 1, 3, and 4 were left practically empty.

From a professional sports bettors perspective, this document clustering algorithm is not completely useless. Cluster 2 was able to group together 55.6% of the entire relevant articles in the corpus. This cluster was made up of 80.1% relevant articles with a sample size of 191 documents. Combining clusters 1, 3, 4, and 5 together, there are 145 documents and only 10.3% are relevant to a sports bettor. These four clusters would make a reasonable filter to remove irrelevant news articles.

4.3. Activity 3: Multidimensional Scaling (t-SNE)

Multidimensional Scaling was accomplished by using the t-distributed stochastic neighbor embedding (t-SNE) algorithm by way of the SKLearn package (SKLearn, 2020b) and visualized with the Yellowbrick library in Python (Yellowbrick, 2020).

4.3.1. Analyst Judgment (t-SNE)

Figure 2 provides a two-dimensional visualization of the six clusters created from Activity 2 for the Analyst Judgement document vectorization methodology. As can be seen, the majority of the documents are spread out across the visual in the second cluster (light green) covering the upper left quarter of the graph or else in the first cluster (dark blue) covering the bottom right quarter of the graph.

Cluster 6 (light blue) documents create a very tight cluster in the bottom right corner. These documents are predominately relevant to sports bettors and it is not surprising they create the most discernible cluster in the model.

The clustering results from this unsupervised learning model were not balanced creating very few distinguishable groupings in the t-SNE visualization.

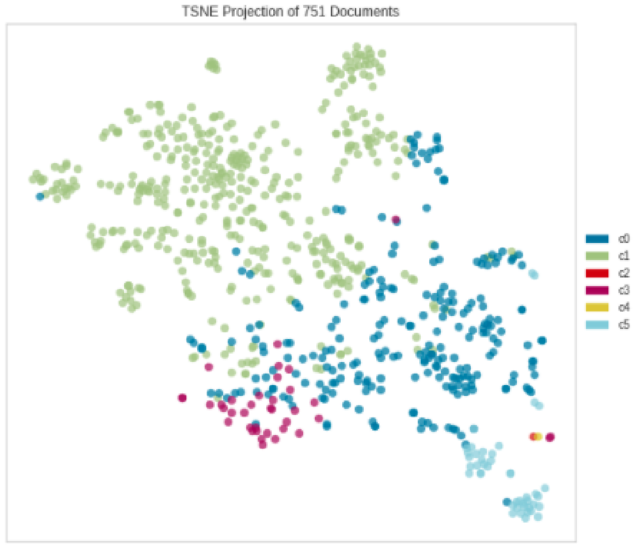

Figure 2: Analyst t-SNE

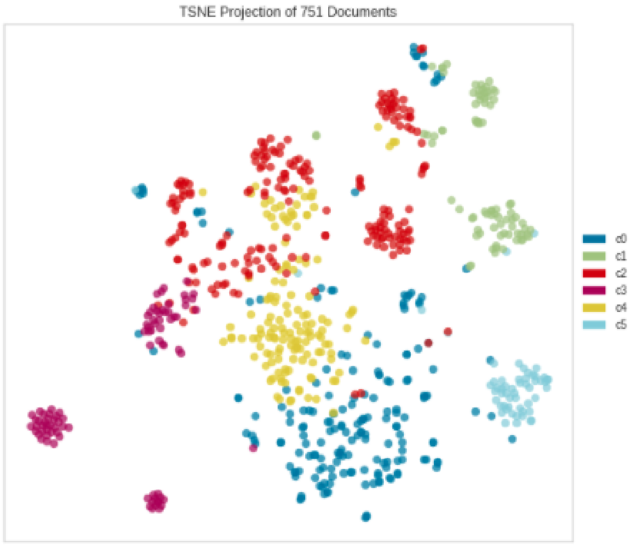

Figure 3: TF-IDF t-SNE

4.3.2. TF-IDF (t-SNE)

Figure 3 provides the two-dimensional visualization of the six clusters created from Activity 2 for the TF-IDF document vectorization methodology. Compared to the other two vectorization methodologies, the TF-IDF t-SNE visualization produces several tightly clustered documents signifying strong differences between the documents in the groupings.

Compared to the other two vectorization methodologies, the TF-IDF t-SNE visualization produces several tightly clustered documents signifying strong differences between the documents in the groups. Cluster 2 ("c1" light green) and cluster 3 ("c2" red) in the upper right quarter are relevant articles with different topics creating several tight clusters.

4.3.3. Doc2Vec Embedding (t-SNE)

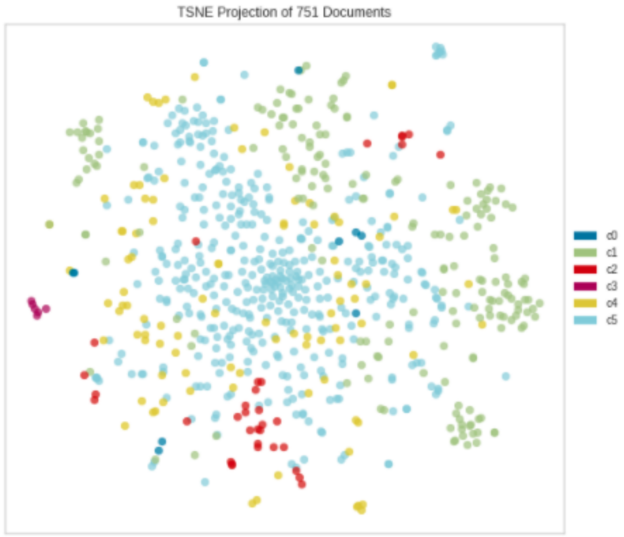

Figure 4: Doc2Vec t-SNE

Figure 4 provides a two-dimensional visualization of the six clusters created from Activity 2 for the Doc2Vec embedding document vectorization methodology.

Cluster 2 ("c1" or the light green color) is interesting to note in this visualization. This cluster contains the majority of relevant news articles for a sports bettor. These relevant articles cover several different topics like trades, pre-game analysis, post-game analysis, and team perception/sentiment. There are several tight green clusters in various places around the outside perimeter of the visual. Each of these tight circles represent the different topics that are all categorized as relevant news articles in the corpus.

4.4. Activity 4: Comparison

The K-Means clustering algorithm and t-SNE visualization with the TF-IDF document vectors performed the best out of the three vectorization methodologies. There was clear discrimination between many of the clusters when using the TF-IDF vectors. The Doc2Vec and Analyst Judgment document vectors resulted in heavily skewed distributions of documents in the clusters. There was also very little discrimination between clusters when visualized with t-SNE.

Focusing on the professional sports bettor's problem to sort between the relevant and irrelevant news articles, the TF-IDF and the Doc2Vec clustering models both provide value. The TF-IDF model was able to find value with each of the six clusters, extracting great value in four of the six clusters. The Doc2Vec model was not able to discern value across all of the documents but the cluster 2 stuck out as a strong positive group.

4.5. Activity 5: Terms t-SNE

The Doc2Vec embedding vectorization method does not directly relate to identifiable terms the way that the analyst judgment and TF-IDF mothodologies do and will be disregarded in Activity 5, 6, 7, and 8.

The document vectors were transposed in order to use the terms as objects instead of the documents. This creates a matrix with 300 rows, one for each term, and 751 columns, one for each document.

4.5.1. Analyst Judgment Terms (t-SNE)

Figure 5 provides a two-dimensional visualization of the ten clusters for the terms used by the Analyst Judgement vectorization methodology. There are no tightly clustered groups in this model.

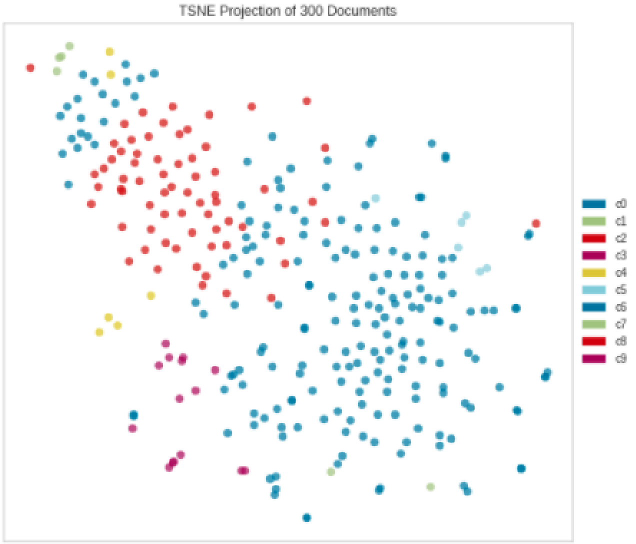

Figure 5: Analyst t-SNE

4.5.2. TF-IDF Terms (t-SNE)

Figure 6 provides a two-dimensional visualization of the ten clusters for the terms used by the TF-IDF vectorization methodology. There are no tightly clustered groups in this model.

4.6. Activity 6: Hierarchical Clustering

The visuals for the Hierarchical clusters, both Analyst Judgment and TF-IDF, are difficult to view because of the size but are included the Appendix in Figure 18 and 19 respectively. An easier to read version is provided in the Google Colaboratory notebook under the Activity 6 header.

4.6.1. Analyst Judgment

Hierarchical clustering using the vectors from the Analyst Judgment approach created three high-level clusters and sixteen clusters within a distance of 100. The dendrogram visualization in Figure 18 shows these clusters.

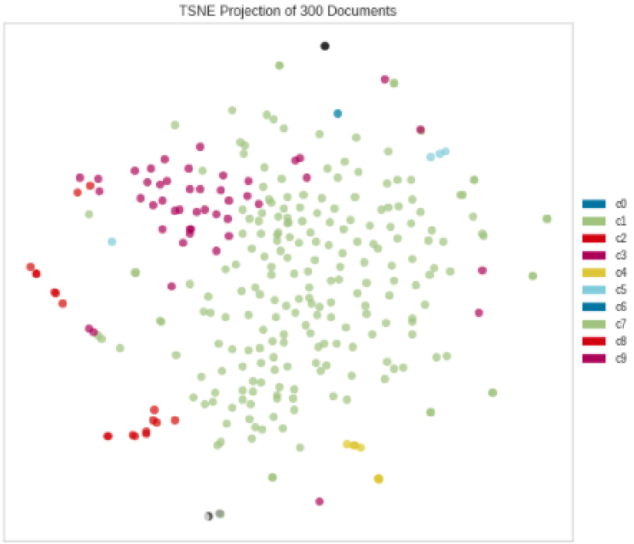

Figure 6: TF-IDF t-SNE

The clustering and ordering of terms in this model make logical sense, for the most part. Equivalence class terms were linked together with very short distances like "Lillard" and "Damian Lillard". Adjacent to Damian Lillard was his teammate McCollum. Days of the week were also grouped right next to each other.

4.6.2. TF-IDF

Hierarchical clustering using the vectors from the TF-IDF approach created seven high-level clusters and nineteen clusters within a distance of 3 (scaling is different than Analyst Judgment).

It was promising seeing equivalence classes in the same clusters or very close together. "Trail blazers", "trail, "blazers", and "portland" are in their own black cluster. Adjacent to this black cluster, a yellow cluster containing the terms "Atlanta", "hawks", and "Atlanta hawks". Days of the week were right next to each other. "Coach", "head", "Head Coach", and "players" were grouped together.

The green cluster contained many of the key words in the relevant articles like "last night", "shoot" "three" "score" "rebound" "assist" "points" average point" "per game" and "win". The red cluster was focused on words that had to do with community outreach and the family.

4.7. Activity 7: Comparison

The t-SNE methodology was much more difficult to interpret compared to the hierarchical clsutering methodology. It appeared as if there were not many distinct clusters for the terms in the corpus using the t-SNE method for both TF-IDF and Analyst Judgment vectors. Compared to the documents as objects clustering t-SNE visualizations in Activity 4, the terms as objects clustering t-SNE visualizations do not provide much value when trying to understand the corpus deeper.

Using a dendrogram to visualize the hierarchical clusters created made for an extremely easy interpretation. The terms that were closer together had higher similarity metrics. Cleveland, Nuggets, Rockets, Houston, Clippers, and Lakers, were all right next to each other in the orange cluster in Figure 18, while Trail Blazers, Trail, Blazers, Portland, and Lillard were all close together in the sky blue cluster.

4.8. Activity 8: Ontology

Figure 7 provides a high-level view of the terms in the NBA news article corpus relevant to a professional sports bettor. There are three categories that the NBA can be broken into from a sports betting perspective: Teams, Games, and Officials.

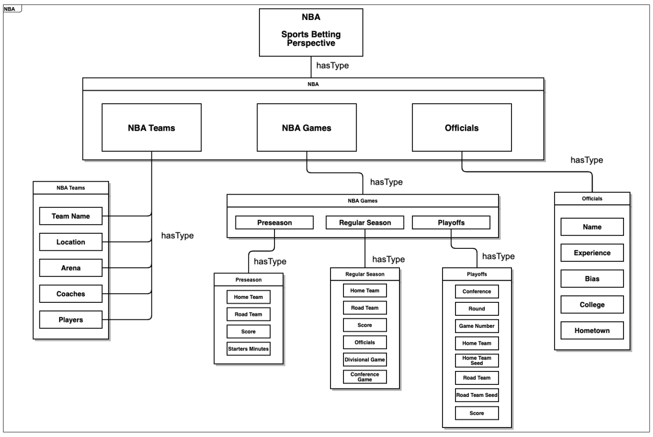

Figure 7: Ontology for the NBA from a Sports Betting perspective

The Golden State Warriors will be used as an example for the NBA team node. NBA teams are comprised of several elements. Each team has its name. Within the name category, there is the actual team name and various nicknames. "Golden State Warriors", "Golden State", "Warriors", "Dubs", "The Town" are five terms that an algorithm would view as different but one with domain knowledge understands they all represent the exact same thing. These terms should be grouped together into a single term to reduce the complexity and noise in the model.

Then each team has a Location. "City by the bay", "Golden State", "California", "San Francisco", "Oakland", "Bayside", "Dub Nation", "Mission Bay" are all terms that represent where the Golden State Warriors are located. Each team also has an arena. "Oakland Arena", "Oracle Arena", "Oracle", "Warriors Arena", and "Chase Center" all represent where the home games for the Warriors are played. The location of the team and the arena they play in for most purposes represent the exact same thing. Each of the 13 terms listed above could easily be grouped together into a single term further reducing the model complexity and removing unneeded noise.

Players and coaches are all people in different positions first and last names and most likely a nickname that are equivalence classes.

From a high-level unsupervised document clustering modeling perspective, the various detailed terms listed above introduce too much noise in the data and should be condensed to the nodes presented in Figure 7.

4.9. Activity 9: Topic Modeling

Independent of the previous activities and the various document vectors, the topic modeling process is another unsupervised learning method used to understand and sort the various NBA news articles in the corpus.

The topic modeling algorithm was fed unigrams, bigrams, and trigrams creating thousands of possible terms to learn from. There were [1]four unique topics created by using the LDA algorithm. Figure 8 provides a word cloud for each topic created in this activity giving a better idea on the types of words in each topic.

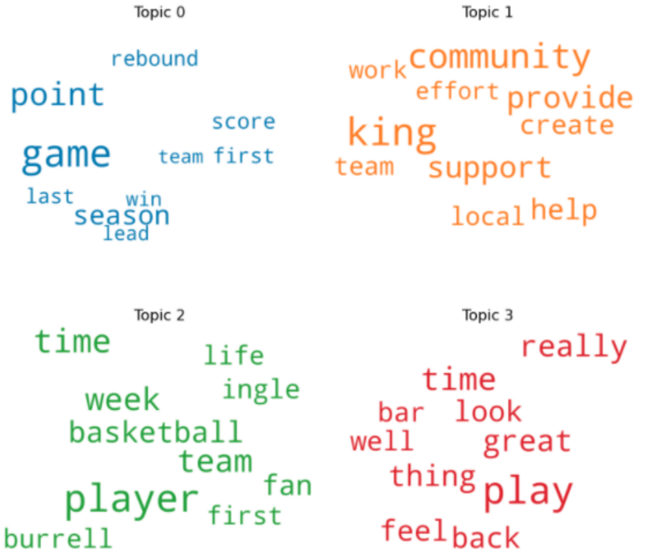

Figure 8: Topic Word Count

Figure 9 and Table 4 show the distribution of documents classified by their dominant topic. The LDA topic model was able to keep a fairly equal number of articles in each topic. This unsupervised learning model was able to avoid the issues of imbalanced, or skewed, distributions see in Activities 2 and 5.

Topics 2 and 3 could be used to filter out irrelevant NBA news articles for a sports bettor. They contain 7% and 20% relevant articles for each topic and 3.6% and 10.9% of the total number of relevant documents in the corpus. Simply filtering out these two topics, the percentage of relevant articles in the corpus increases from 36.6% to 51.4%.

[1] Python is a base 0 language meaning that 0 is the first term, which could make some of the visuals confusing. "Topic 0" is the first topic, while "Topic 3" is the fourth and final topic created in this activity.

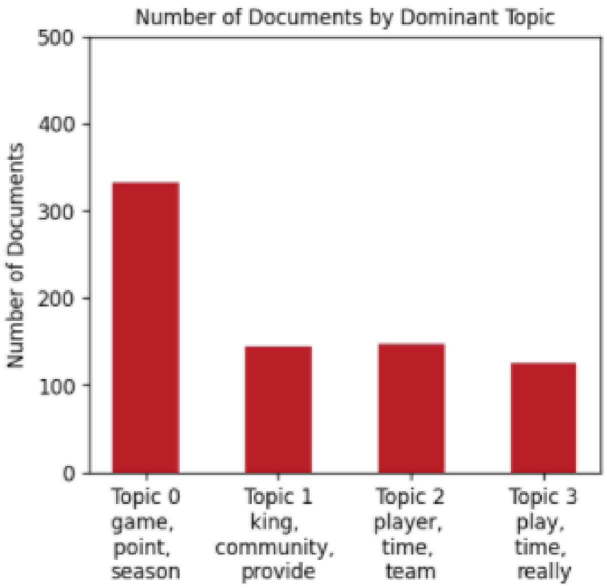

Figure 9: Topic Document Counts

	Topic 1	Topic 2	Topic 3	Topic 4
Relevant	192	10	30	43
Irrelevant	140	135	119	82
Total	332	145	149	125

Table 4: NBA News Articles per topic.

4.9.1. Topic 1: Game Analysis

The first topic created in the LDA modeling contained the majority of the articles that are relevant for a sports bettor. This topic focuses on words that relate to pre-game and post-game analysis with words like "Game", "Point", "Score", and "Win".

The LDA model's equation for Topic 1 is:

$$\begin{aligned} Topic\ 1 &= 0.062 * Game + 0.042 * Point \\ &+ 0.029 * Season + 0.017 * Score \\ &+ 0.016 * Rebound + 0.016 * First \\ &+ 0.016 * Last + 0.015 * Win \\ &+ 0.014 * Lead + 0.013 * Team \end{aligned}$$

Figure 10 is an interesting visual showing the weights of the most important terms used in this topic as well as the counts of those terms. Figure 11 contains a list of the top 30 words used within this topic.

4.9.2. Topic 2: Community Service

The second topic that was created focused almost entirely on news articles dealing with community outreach. This topic could be used to filter out the irrelevant NBA news articles for sports bettors.

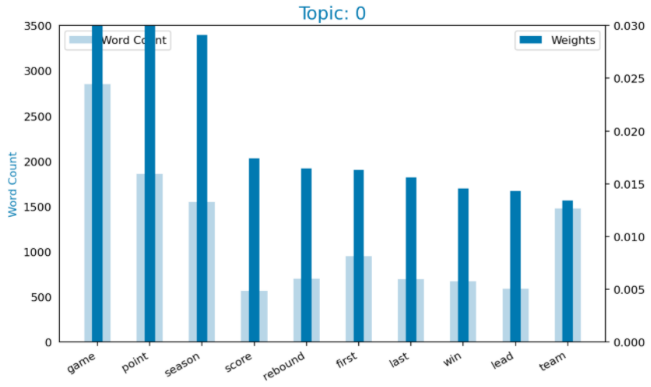

Figure 10: Topic 1 Word Weights

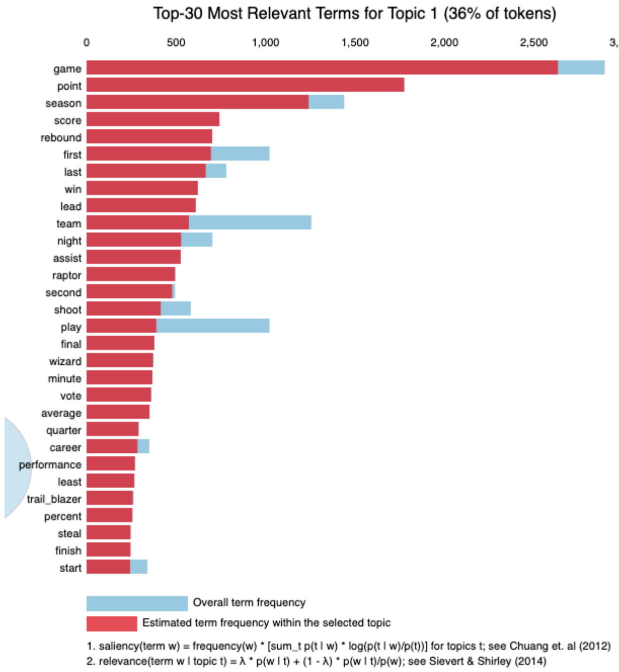

Figure 11: Top 30 words in Topic 1

The LDA model's equation for Topic 2 is:

$$\begin{aligned} Topic\ 2 &= 0.022 * King + 0.018 * Community \\ &+ 0.014 * Provide + 0.013 * Support \\ &+ 0.012 * Help + 0.010 * Create \\ &+ 0.009 * Team + 0.009 * Local \\ &+ 0.009 * Work + 0.008 * Effort \end{aligned}$$

The NBA has a strong relationship with the communities the teams are in and these articles contained key words like "Family", "Community" "Provide" Support" "Help" "Local" "Team Work" or "Program". Figure 13 contains a list of the top 30 words used within this topic and Figure 12 shows the weights associated with the top words.

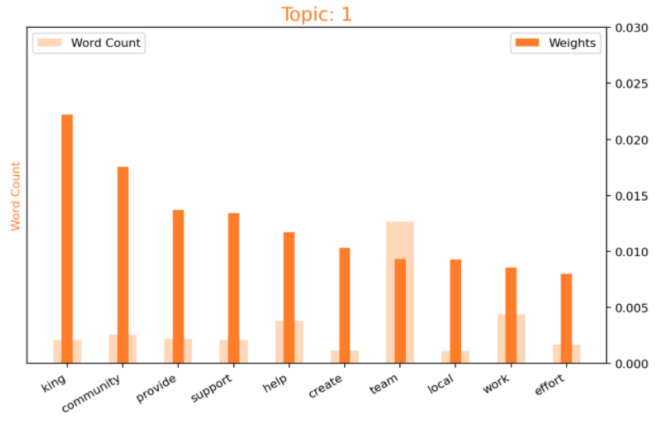

Figure 12: Topic 2 Word Weights

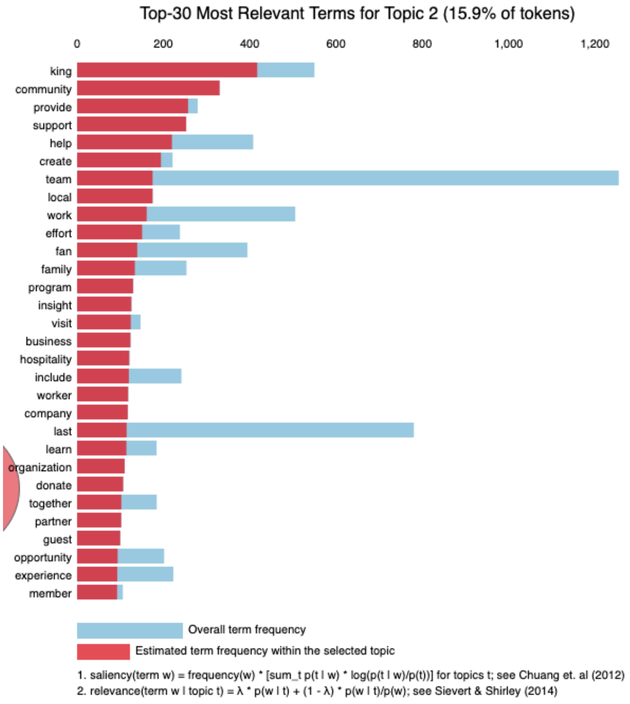

Figure 13: Top 30 words in Topic 2

4.9.3. Topic 3: Player Focused

Topic 3 is focused on players as people and not superstars. These articles discuss what the NBA players do outside of competing on the hardwood.

The LDA model's equation for Topic 3 is:

$$\begin{aligned} Topic\ 3 = 0.021 * Player + 0.015 * Time \\ +0.014 * Team + 0.014 * Week \\ +0.012 * Basketball + 0.011 * Fan \\ +0.010 * Ingle + 0.010 * Burrell \\ +0.009 * Life + 0.009 * First \end{aligned}$$

Figure 14 is an interesting visual showing the weights of the most important terms used in this topic as well as the counts of those terms.

Figure 15 contains a list of the top 30 words used within this topic.

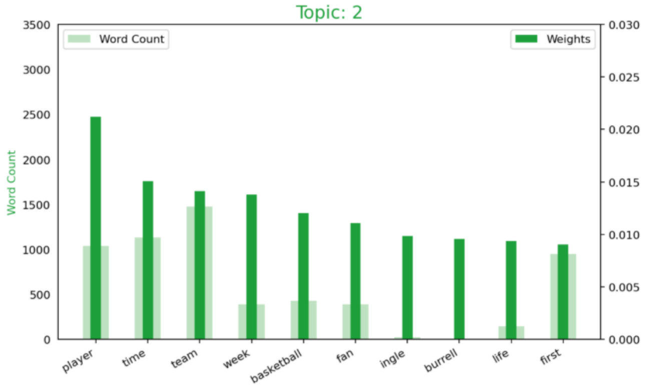

Figure 14: Topic 3 Word Weights

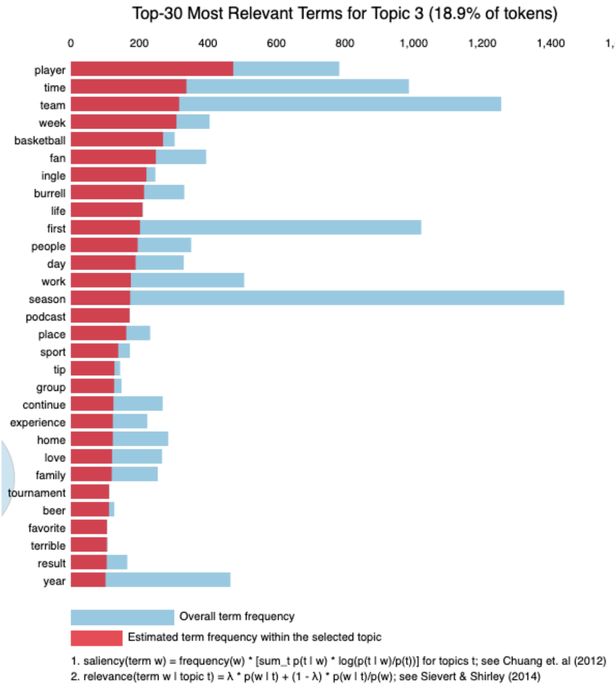

Figure 15: Top 30 words in Topic 3

4.9.4. Topic 4: Covid Shutdown (looking back)

This topic includes many of the miscellaneous articles, especially those written after the NBA shut down because of the Covid-19 virus. This topic also includes many of the trade specific relevant articles.

The LDA model's equation for Topic 4 is:

$$\begin{aligned} Topic\ 4 = 0.018 * Play + 0.012 * Time \\ +0.011 * Really + 0.011 * Great \\ +0.010 * Thing + 0.010 * Look \\ +0.010 * Back + 0.010 * Feel \\ +0.009 * Well + 0.008 * Bar \end{aligned}$$

Figure 16 and Figure 17 provide graphs of the key terms used in this topic and their associated weights.

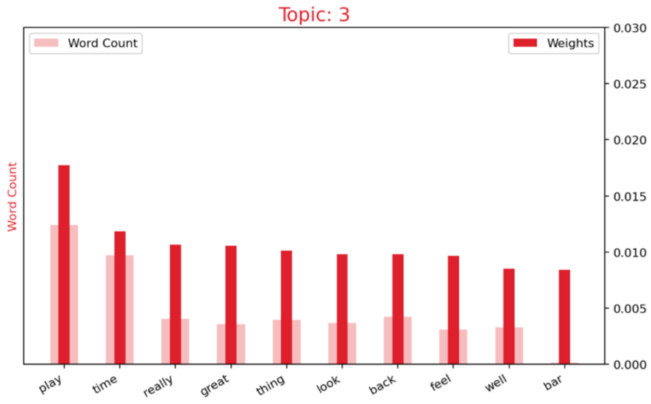

Figure 16: Topic 4 Word Weights

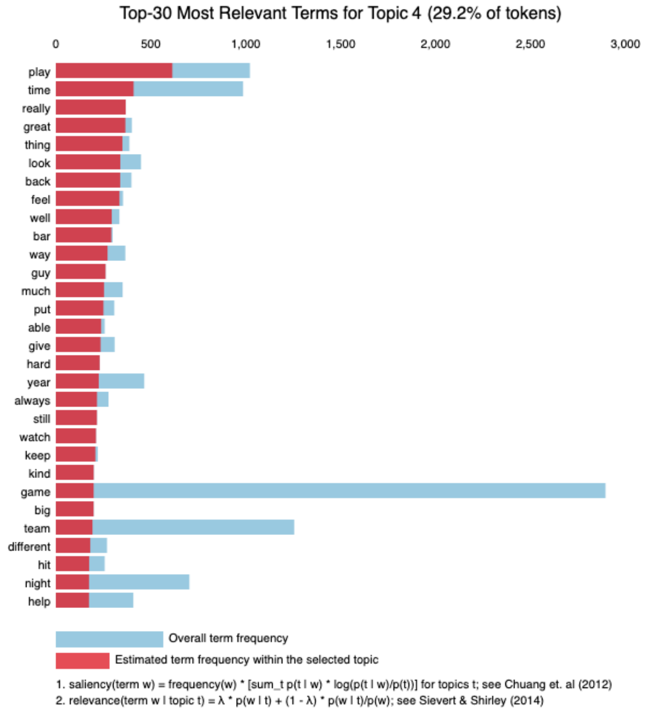

Figure 17: Top 30 words in Topic 4

5. Discussion and Conclusions

There are many benefits to using unsupervised machine learning techniques. However, they can produce uncontrollable and often sub-optimal results depending on the data and the use case. For the purpose of a professional sports bettor overwhelmed with incoming news articles, the unsupervised machine learning methods of clustering and topic modeling performed worse than the document classification model built in the previous project (Lee, 2020a).

There is potential to achieve even more efficiency for a professional sports bettor by applying supervised learning methods and unsupervised learning methods in tandem. Future work could examine the accuracy and effectiveness of a data flow pipeline utilizing document filtering based on the supervised classification model followed by an unsupervised topic modeling algorithm to sort the remaining relevant news articles. This flow may prove to be the optimal solution to the original problem posed in this and the previous NLP project that professional sports bettors face.

As for the unsupervised machine learning results, future work should include improvements to the pre-processing of the raw text data. For example, the clustering and topic modeling algorithms were erroneously creating divisions based on which team the news article was discussing. This is problematic and defeats the purpose of using these techniques owing to the knowledge that each article is tagged with the team based on which team website the article was scraped from.

A potential solution could come by building off of the ontology in Activity 8 with raw terms grouped together at the given node from Figure 7. Each team name mentioned in the articles could be grouped together into a single "NBA Team" term allowing for an anonymized term representing all teams to stop clusters from forming around a specific team. This could also be extended to player names; a single term can be used to represent all NBA players. Depending on the situation, it may work better to have NBA players be represented by two terms, a term for the starting players on a team and a term for the bench players on a team.

There were many lessons learned and extensions to future work that have been unearthed during this study. This future work will need to be completed to better calibrate the document clustering models before A.I. Sports will be able to confidently implement the models as a product in the real sports betting world.

In conclusion, while there needs to be more work to improve the results, this study was still able to accomplish the primary goal of using unsupervised machine learning techniques to better understand and organize the NBA news article corpus. Unsupervised learning algorithms provide a quick solution for understanding large quantities of text documents. Valuable insights into the NBA news article corpus were gained both by using the documents as objects and by using the terms in each document as objects through this study.

Appendix

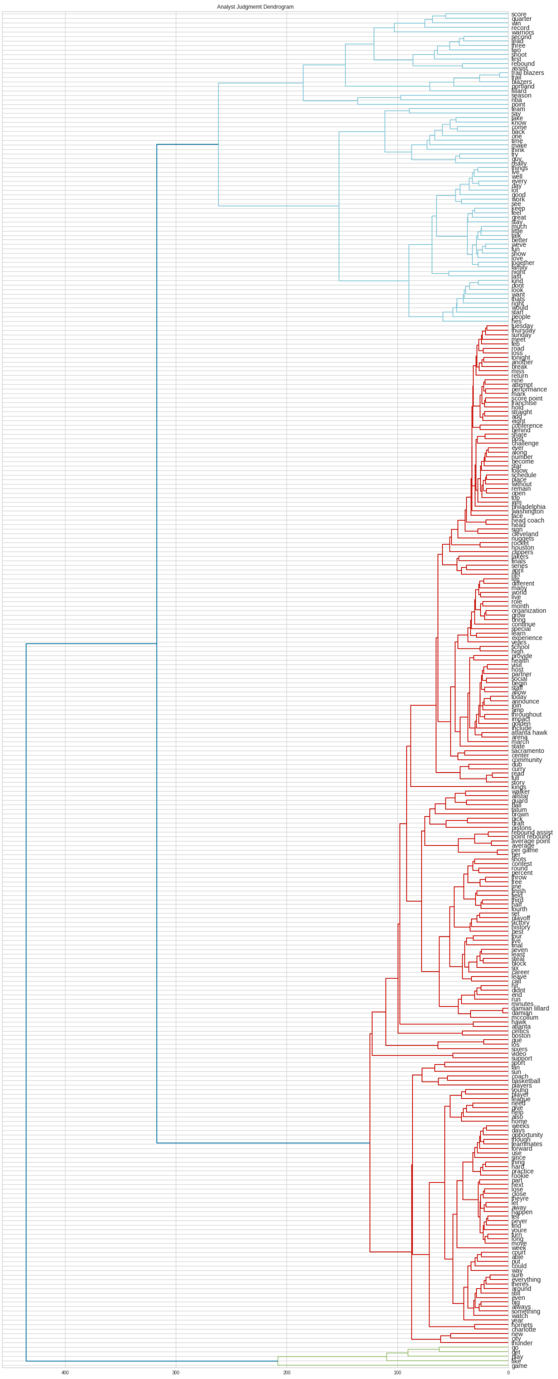

Figure 18: Analyst Dendrogram

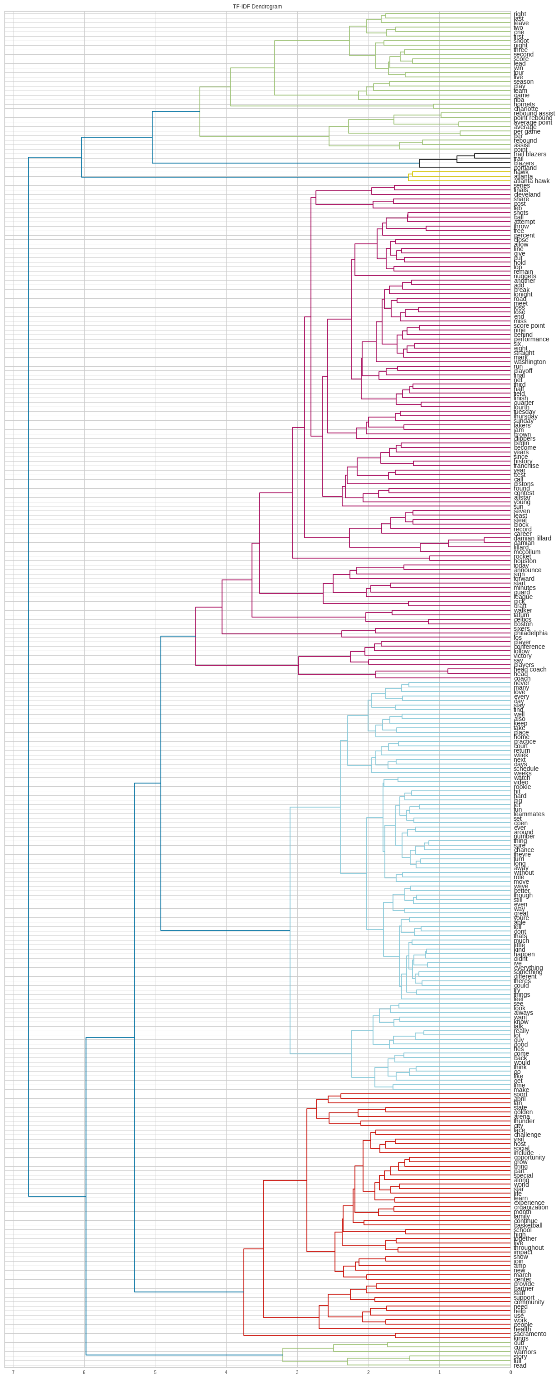

Figure 19: TF-IDF Dendrogram

References

Albon, C., 2018. Python machine learning cookbook: Practical solutions from preprocessing to deep learning. O'Reilly Media, Inc., Sebastopol, CA.

Huynh, P., 2020. Reduced reuters cluster analysis. https://canvas.northwestern.edu/courses/112789/pages/reduced-reuters-cluster-analysis-code-paul-huynh?module_item_id=1425574. Accessed on 2020-05-15.

Izenman, A.J., 2008. Modern multivariate statistical techniques: Regression, classification, and manifold learning. Springer, New York: NY.

Lane, H., Howard, C., Hapke, H.M., 2019. Natural Language Processing In Action. Manning Publications Co., Shelter Island, NY.

Lee, J., 2020a. Document classification model: Nba news articles. URL: https://github.com/papagorgio23/NBA_News_Spiders/blob/master/Classification%20Model%20Project.pdf.

Lee, J., 2020b. Focused web crawler: Nba team specific news articles. URL: https://github.com/papagorgio23/NBA_News_Spiders/blob/master/Focus%20Web%20Crawler%20Project.pdf.

Lee, J., Shephard, I., Wolande, P., 2018. A.I. Sports. https://aisportsfirm.com/.

Patel, A.A., 2019. Hands-on unsupervised learning using Python: How to build applied machine learning solutions from unlabeled data. O'Reilly Media, Inc., Sebastopol, CA.

Prabhakaran, S., 2020. Topic modeling visualization: How to present the results of lda models. https://www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/. Accessed on 2020-05-15.

Řehůřek, R., Sojka, P., 2010. Software Framework for Topic Modelling with Large Corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA, Valletta, Malta. pp. 45–50. http://is.muni.cz/publication/884893/en.

SKLearn, 2020a. Countvectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html. Accessed on 2020-05-01.

SKLearn, 2020b. Sklearn manifold tsne. https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html. Accessed on 2020-05-15.

SKLearn, 2020c. Tfidfvectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. Accessed on 2020-05-01.

Van Der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. Journal of Machine Learning Research 9, 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Yellowbrick, 2020. t-sne corpus visualization. https://www.scikit-yb.org/en/latest/api/text/tsne.html. Accessed on 2020-05-15.