The Analytics Edge, Final Report:
Building an Article Recommender for Bleacher Report Basketball

for Professor Dimitris Bertsimas and Emma Gibson

by Stephen Albro & Cyrille Combettes
salbro@mit.edu, cyrille@mit.edu

May 7, 2018

1 Abstract

2 Motivation and Project Abstract

As avid basketball fans, we have enjoyed reading a variety of NBA articles to keep up-to-date on the league during our busy year in the MBAn program. Bleacher Report, a primary digital destination for sports article readers, contains well written articles covering the league, and competes as advertisement real-estate with the likes of ESPN.com, FoxSports.com, CBSSports.com, SBNation.com, and others.

In an age of cut-throat digital competition, it is vital for websites to be able to retain visitors. The hope for a Bleacher Report article reader is two-fold. First, that the visitor jumps from article to article *on the website*, rather than returning to the search engine results page. Second, that the visitor falls in love with the content and coverage of Bleacher Report's articles, and develops a loyalty to the website.

We feel that having a micro targetted article recommendation system would provide Bleacher Report with an analytics-based edge over its competitors, since well targeted articles could increase the chances of prolonged visits, and increase website loyalty by making users feel known. In this paper, we develop a prototype for an article recommendation system using state of the art natural language processing techniques for both cleaning and analyzing bleacher report articles. Specifically, we aim to cluster over 3000 Bleacher Report articles by testing three different approaches: Topic Modeling, Word2Vec text embeddings, and NMF (CYRILLE WRITE OUT FULL NAME AND TALK ABUOT WHAT NMF IS). {GO INTO MORE DETAIL HERE ABOUT HOW CLUSTER HELPS THE RECOMMENDER. }

3 Data Collection and Cleaning

3.1 Data Collection

Bleacher Report does not provide easy access to their articles in a public database. However, News API is a company that provides API access to major news articles from a variety of sources, and this access is free for non-commercial projects. Using NewsAPI, we queried Bleacher Report NBA articles from October 17, the start of the 2017-2018 NBA season. Our entire query consisted of joining words like basketball and NBA with all 30 teams and the top 25 active players. In the end, we received 3314 URLs (with their corresponding authors, titles, and other metadata) in return. We then wrote a script to request the webpage for each URL and scrape its content, picking the paragraphs out of the HTML.

3.2 Data Cleaning

In the real world, data is never clean, and this is magnified in the case of textual data. There were a number of preprocessing steps we had to do before our articles were ready for analysis. In our cleaning efforts, we made decisions *based on the purpose of our task* - to cluster documents for better recommendation power. In particular, a bag-of-words approach to cleaning was preferred over one that preserved punctuation and grammar.

3.2.1 Punctuation Removal, Initial Stopwords, and Lowercasing

First, we tokenized each article. In natural language processing, tokenizing is a way of chopping up a document into pieces, the most obvious way being to tokenize by *word*, which is what we initially did. Next we discarded any non-alphabetical tokens (e.g. 8pm, 704, !, .), and converted every token to lower case. After converting everything to lower case, we did our first round of stop word removal (we remove more later). In NLP a stop word is any word that you remove because its insignificant for your application. In this first round, we removed basic articles (the, a) and about 170 other words deemed insignificant by Python's Natural Language Toolkit, mostly personal pronouns and simple verbs.

3.2.2 Specialized NBA Tokenization

Next, since we were analyzing NBA articles, we had to take into account the fact that each player (and each team for that matter) can be referred to in a variety of ways. For example, the NBA superstar LeBron James can be called LeBron James, LeBron, Bron, or James, and the Boston Celtics might be formally referred to as such in the title, but in the article's body they might just be Boston. For this reason, we compiled lists of team names, players, and coaches. For each article, we replaced all forms of each entity with its underscored full version, so that for example LeBron James would always be lebron_james, no matter if he occurred as Bron, James, or Lebron in a given sentence. At times it was necessary to infer from ambiguous usage. We concluded that, for

example, the word *Boston* should be replaced by boston_celtics only if the entire phrase Boston Celtics appeared somewhere in the article. This specialized, application-specific tokenization step allowed us to capture much more information about each superstar/team/coach than would for example, treating Bron as a separate player as Lebron.

3.2.3 Trimming Down the Vocabulary

A model is only as good as its input, and so as a last step, we realized that we needed to trim down the vocabulary of our articles. In NLP, a *vocabulary* is simply the complete set of words that the model knows about. We already removed basic stop words, but figured that many more could be removed. For clustering documents, its important to identify information-containing words (player and team names, high-impact verbs, league terminology), and less important to identify mundane nouns and verbs (table, cup, gave, took) even if their not considered stop words.

To identify our functional vocabulary, we used a *document frequency* approach. The document frequency of a given word is simply the fraction of documents (in this case articles) that the word appears in. The word *the*, for example, has a document frequency of 1.0. If a word occurs too *infrequently*, it is probably a unique name or an article-specific entity, and thus unhelpful in forming clusters. On the other hand, if a word occurs too *frequently*, it is probably a common word and thus contains very little information related to topics.

Thus, we sought lower and upper document frequency cutoffs for including a word in the vocabulary. As a sanity check, we knew our upper cutoff had to be at least high than the document frequency of LeBron James (0.22, he the most famous superstar), since every player should be in our vocabulary. After seeing irrelevant words present with an upper cutoff of even 0.5, we knew that the upper cutoff had to be lower than that, and decided upon 0.35 after experimentation (at this point the eye test was possible). Our lower cutoff was 0.01, which means that if a word appeared in less than 34 out of 3314 total articles, it was discarded. In total, the new vocabulary consisted of about 3500 tokens with document frequencies within the range (0.01, 0.35), including the major players and teams, and, we think, the important, topic-filled sports and NBA terminology. To give a sense, some of the first few tokens printed from our vocab set include *son, statistics, honor, turnover, athleticism, pulled, dribble, andrew_wiggins, offseasons, attitude, cuts,* and *jump.*

4 Methods

- include tuning and cross validation, etc.
- include tables, cool visualizations

4.1 ¡METHOD 1¿

4.2 ¡METHOD 2¿

4.3 ¡METHOD N¿

5 Discussion

5.1 Which Approach is Best

- how to we measure success, etc.

5.2 Why This Project Was Important

- we learned a lot about all three methods - most of the world's data is text
- lots of online advertising
- lot of research in NLP because anlytics provides an edge