Properly Handling Proper Nouns: Equivalence Classes or Removal?

Christopher J. Rico

Northwestern University

Introduction and Problem Statement

Document vectorization is an unsupervised natural language processing (NLP) approach that encodes an entire corpus of documents as hyperdimensional vectors. These vectors can then be used to create “clusters” of documents that are most alike using an unsupervised machine learning method called k-means clustering.

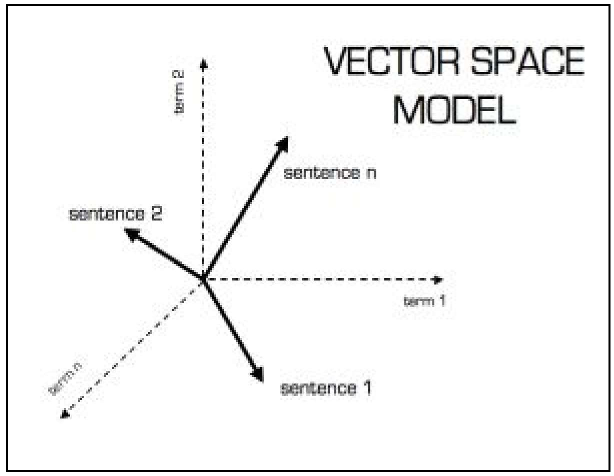

Fig 1 - Various sentences represented as vectors

The corpus in question consists of film reviews covering a broad swath of films. Within a corpus this broad, the common important terms among documents that review the same film (or same film series) are likely to be the characters, actors, director, and the film title itself. However, this presents a problem: we are interested in employing NLP on these documents to understand how they relate and differ from one another on a level deeper than this. If we wish to do more than simply understand which documents mention the same people, it may be of value to examine how different proper noun handling methodologies affect clustering outcomes.

This analysis attempts to evaluate the effect of 2 different proper noun handling techniques on k-means document clustering. One method will be to simply remove proper nouns altogether. The other method will be to condense the range of proper nouns down into movie-centric “equivalence classes” that will form the beginnings of a cinematic knowledge ontology. This is a first attempt at engineering contextual knowledge to be used by the Doc2Vec NLP algorithm during document vectorization.

Dataset

The dataset consists of a single corpus of 61 documents, each contributed by a student in the course. Within the corpus, each document is a film review written in English, truncated to roughly 500 words by the student who submitted it.

Manipulation of the corpus was performed in Google Colaboratory (a cloud-based Jupyter notebook) running a Python 3 interpreter. A variety of Python functions and NLP libraries were used to prepare the data for term extraction and document vectorization. Each document was first tokenized and stripped of punctuation and non-alphabetical characters. Next, tokens shorter than 4 characters were removed, as well as any stopwords in the Natural Language Toolkit (NLTK) English stopwords list. Finally, all characters were converted to lowercase.

Research Design and Methods

This experiment was run using the provided code in the environment described above. Various industry-standard NLP libraries were used to perform the term extraction and document vectorization: Computing k-means clusters was performed using SciKit-Learn's Kmeans library. Documents vectorized by Gensim Doc2Vec will be grouped into 8 clusters with one of two treatments: either proper nouns removed completely (PNR), or proper nouns condensed (PNC) into ontologically significant equivalence classes. Cluster composition and mean cosine difference per cluster will then be compared to clusters constructed with proper nouns included (PNI) to gain insight into how proper noun handling affects clustering of the corpus.

Proper nouns were discovered iteratively by performing k-means TF-IDF extraction/clustering on the corpus, and simply looking at the 'most important' terms within each cluster. Selected proper nouns were then added to an auxiliary stopwords list (see Appendix 1), and removed during corpus pre-processing. In general, TF-IDF extraction on this corpus tended to favor proper nouns as the most important terms, so this was an effective way to remove those which had the most influence on term extraction and (presumably) document vectorization.

Equivalence classes were constructed by extracting the proper nouns from each document. Each movie franchise was then assigned a list of its proper nouns, to be replaced by the title of the film. Equivalent terms were then converted to their assigned EC prior to document tokenization. For a complete list of ECs and their included terms, see Appendix 2.

Results

Both proper noun removal and proper noun condensation had substantial effects on cluster composition. Compared to clusters constructed with proper nouns included, PNR clusters were more heterogeneous with respect to film title and topic. PNR clusters also had much more uneven numbers of films per cluster. PNC clusters showed a similar effect of having uneven numbers of films per cluster, but clusters made more "sense" in terms of grouping based on film title and topic.

Both PNR and PNC clusters had a mean cosine distance that was less than that of PNI clusters, with PNC clusters having the lowest mean cosine distance of all three treatments.

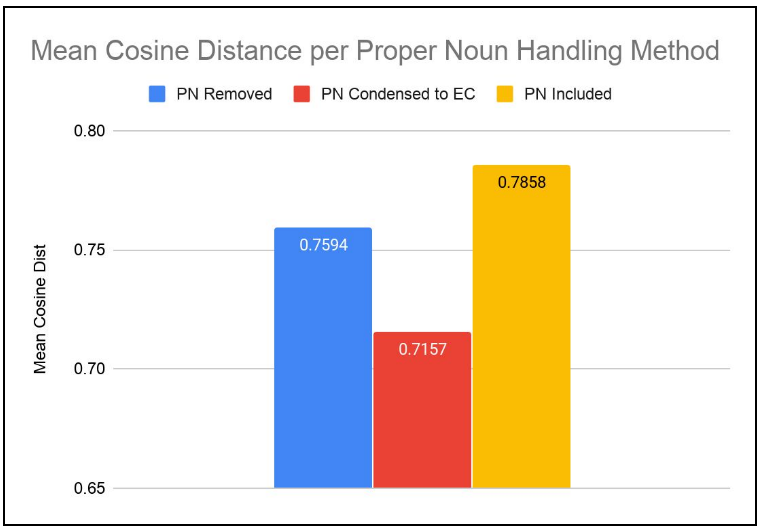

Fig. 2 – Mean cosine distances for PNI, PNR, and PNC clusters

Analysis

Cosine Difference per Treatment

Both methods of handling proper nouns within the corpus led to lower mean cosine distances between documents in each cluster. Initially, I took this result to mean that handling proper nouns in one way or another led to clusters that were composed of more alike documents. However, upon closer examination of the mean cosine distance of individual clusters for each treatment, higher number of films per cluster seemed to be correlated with higher cosine differences. This makes sense, as large clusters are likely to cover more "ground" in terms of document vector space. Because clustering after PNR and PNC tends to result in many smaller clusters and one or two massive clusters, the *mean* cosine difference for each of these treatments appears smaller than that of PNI clusters, which tend to be more homogeneous in size. In fact, if we remove the effect of cluster size by taking a weighted average of cosine difference per treatment, it's clear that cluster size is actually driving much of the differences in mean cosine difference between treatments.

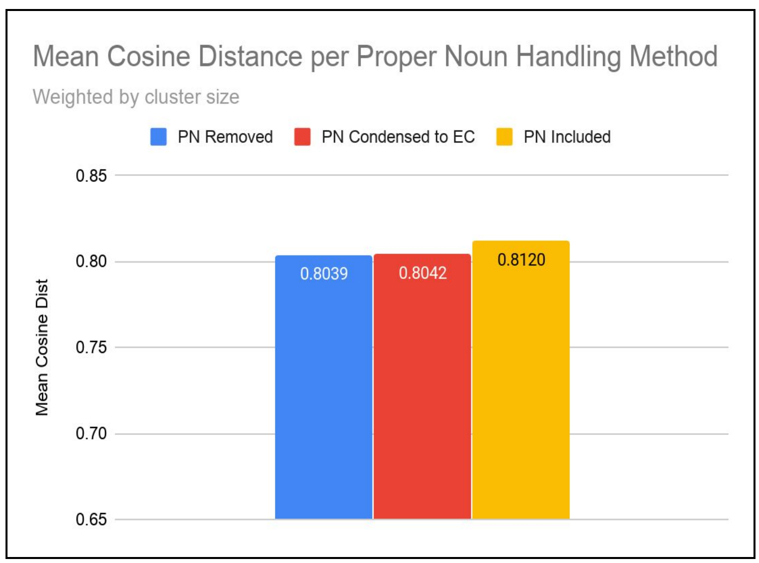

Fig 3 – Mean cosine difference per treatment, weighted by cluster size. Much less variation between treatments is visible.

Cluster Composition

Proper noun handling had odd and unexpected effects on cluster composition. Primarily, both treatments had the effect of pushing clusters to be much more heterogeneous in size – standard deviations of cluster size in PNR and PNC treatments were more than double that of PNI clusters. This translates to both PNR and PNC cluster distributions tending towards several smaller clusters that contain dissimilar film titles, with a single, large cluster that contains similar films that cluster sensibly based on topic. By comparison, PNI clusters tend to be relatively homogeneous in size, but contain documents that review the same films, whether or not the films included are related to one another in topic.

Both proper noun handling methods certainly have stranger and less comprehensible effects on cluster creation than I had anticipated. However, I was surprised to see that the effects of PNR were so similar to those of PNC. Possibly, condensing all proper nouns related to a film down to a single equivalence class is analogous to simply removing pronouns altogether. More time must be spent constructing meaningful equivalence classes that provide contextual knowledge about the corpus to the Doc2Vec algorithm. This may lend the algorithm more generalized knowledge about topics in the corpus, which is extremely valuable.

Conclusions

In conclusion, I was able to observe the effects on both cluster composition and cluster cosine difference resulting from two different methods of handling pronouns. Both methods yielded similar results in that they pushed cluster size to be more variable, but pushed movies with similar topics closer to one another.

In the future, it would be excellent to develop more sophisticated methods for engineering and constructing equivalence classes. While the method described here certainly worked in a pinch, building an ontology of self-referential, contextual concepts needs robust data storage and lookup that may be better implemented as a database or graph than a simple Python dictionary.

Moving forward to a classifier from these results would require developing a more sophisticated tree of equivalence classes to help engineer more contextual knowledge into the corpus before vectorization. Furthermore, we would obviously need to assign agreed-upon classes to each document in the corpus, develop a training and test set, and maybe consider augmenting the training set, as 61 documents is a rather small set to train with.

Clustering vs. Classification

Within the realm of machine learning, there are two broad categories of models: supervised and unsupervised. These two approaches each require different types of data, and generate different types of results from one another (Bisht, 2019). One approach is supervised learning, which requires labeled training data: this is data which has many different features per data point, and an assigned output 'target' that either takes the form of a continuous value or a discrete class. Supervised models (of which there are many) are 'trained' using this training data to learn the mapping function from the input to the output (Brownlee, 2019). Then, when presented with a new input data point, the model can (usually) output either its predicted class or predicted continuous value. Unsupervised learning, on the other hand, is much simpler. Data points are grouped together into a user-defined number of most-alike "clusters". Unlike supervised learning methods, unsupervised methods are not told how to classify these data points – clusters are constructed based on feature similarity without the help of class labels.

References

Bisht, A. (2019, October 3). ML: Classification vs Clustering. Retrieved May 19, 2020, from https://www.geeksforgeeks.org/ml-classification-vs-clustering/

Brownlee, J. (2019, August 12). Supervised and Unsupervised Machine Learning Algorithms. Retrieved May 19, 2020, from https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

Appendix 1. Proper nouns removed from documents

abrams	cobbs	hannah	knight	nolan	scotts
avengers	cooper	harrison	lebowski	nolans	skywalker
batman	deckard	harvey	ledger	pacino	spielberg
blade	denby	hauer	machina	palpatine	stark
blade runner	eckhart	hayward	martian	pesci	tolkien
blank	ex machina	hobbits	marty	preston	tyrell
blank check	fellowship	hoffa	marvel	quigley	victoria
bonnie	forky	hooper	matrix	replicants	walle
bonsall	frank	inception	mcconaughey	ridley	wargames
brand	frodo	infinity	mcfly	rings	woody
bridges	gandalf	interstellar	merkin	runner	zemeckis
caleb	garland	irishman	michael	russel	
check	george	jackson	minority	russell	
christian	gordon	jeff	minority report	scorsese	
christopher	gotham	joker	nathan	scott	

Appendix 2. Equivalence classes built from list of proper nouns

Lord of the Rings

Lord of the rings	gandalf	frodo	hobbits	hobbit
samwise	jackson	peter jackson	baggins	frodo baggins
rings	lord of the rings	elijah	elijah wood	middle earth
mordor	fellowship of the ring			

Dark Knight

dark knight	christian bale	bale	aaron eckhart	eckhart
heath ledger	ledger	gordon	gotham	joker
harvey dent	dent	rachel dawes	dawes	dark knight
knight	harvey	clown		

Blade Runner

blade runner	deckard	harrison	blade	runner
tyrell	ridley scott	ridley	scott	scotts
ridley scotts	hannah	daryl hannah		

The Big Lebowski

the big lebowski	lebowski	bridges	jeff bridges
denby	the dude	duderino	merkin

Interstellar

mcconaughey	cooper	brand

Toy Story

toy story	toy	woody	forky	bonnie

Walle

wall-e	walle	axiom	hello dolly

Avengers

iron man	stark	infinity stone	infinity war
infinity	howard stark	tony stark	russo
superheroes	downey	robert downey jr	

Blank Check

blank	check	blank check
preston	bonsall	quigley

Big Short

big short	mckay	burry	
michael	shipley	short	the big short

Irishman

frank	pesci	pacino	hoffa
russell	scorsese	bufalino	martin scorsese

Star Wars

star wars	skywalker	abrams
force awakens	lightsaber	palpatine

Cats

hooper	hayward	victoria	webber

Ex Machina

ex machina	machina	caleb	nathan	garland

Minority Report

anderton	minority	minority report	spielberg	precrime	witwer

Inception

cobbs	totems

Back to the Future

back to the future	mcfly	zemeckis	marty	marty mcfly	robert zemeckis	brown	doc brown	delorean

War Games

war games	wargames	ripley	lanter

Appendix 3. Clusters with proper nouns included in corpus

0	1	2	3
Big_Lebowski	Blank_Check	Big_Short	Big_Lebowski
Dark_Knight	Cats	Big_Short	Big_Lewbowski
Ex_Machina	Cats	Blank_Check	Dark_Knight
Inception	Inception	Ex_Machina	Interstellar
Star_Wars	Inception	Irishman	Interstellar
Star_Wars	Interstellar	Irishman	LOTR
Toy_Story	Martian	LOTR	Minority_Report
	Minority_Report	LOTR	War_Games
	Toy_Story	Matrix	
	War_Games	Minority_Report	
		Star_Wars	
		Walle	

4	5	6	7
Avengers	Big_Short	Avengers	Blank_Check
Back_Future	Blade_Runner	Back_Future	Interstellar
Blade_Runner	Dark_Knight	Back_Future	Toy_Story
Cats	Walle	Blade_Runner	Walle
Ex_Machina	War_Games	Iron_Man	
Gravity		Matrix	
Irishman		Matrix	

Appendix 4. Clusters with proper nouns removed from corpus

0	1	2	3
Walle	Inception	Blank_Check	Toy_Story
Minority_Report	LOTR	Cats	Back_Future
Ex_Machina	Cats	Avengers	Star_Wars
Dark_Knight	Cats	Matrix	Iron_Man

Big_Short	Star_Wars	Big_Short	Big_Lebowski
	Matrix		Ex_Machina
	Interstellar		War_Games
			Dark_Knight
			Dark_Knight
			Irishman
			Avengers
			Big_Lebowski
			Big_Short
			War_Games
			Irishman
			Blade_Runner
			Toy_Story
			Blank_Check

4	5	6	7
Back_Future	Star_Wars	Irishman	Martian
Blade_Runner	Interstellar	LOTR	War_Games
LOTR	Ex_Machina	Gravity	Minority_Report
Inception	Interstellar	Interstellar	Back_Future
Inception	Toy_Story	Minority_Report	
Matrix		Walle	
Blank_Check		Big_Lewbowski	
Walle		Blade_Runner	

Appendix 5. Clusters with proper nouns condensed into ECs

0	1	2	3
Irishman	Walle	Martian	Star_Wars
Ex_Machina	Toy_Story	Blank_Check	Avengers
War_Games	Back_Future	Cats	Minority_Report

Big_Short	Star_Wars	Gravity	Back_Future
Dark_Knight	Iron_Man	Inception	Toy_Story
LOTR	LOTR	LOTR	Big_Lewbowski
	Interstellar	Cats	
	War_Games	Big_Short	
	Ex_Machina	Inception	
	Minority_Report	Interstellar	
	Big_Short	Star_Wars	
	War_Games	Matrix	
	Matrix	Interstellar	
	Irishman		
	Dark_Knight		
	Blade_Runner		
	Walle		
	Blank_Check		

4	5	6	7
Blade_Runner	Big_Lebowski	Irishman	Back_Future
Avengers	Minority_Report	Toy_Story	Dark_Knight
Big_Lebowski	Ex_Machina		Matrix
Cats	Inception		Interstellar
Blade_Runner			Blank_Check
			Walle