



# Comparing Documents Similarities Using Different Methods

Presented by:

Chris Olen, Ann Eitrheim, Michael Kowalski, Jimei (Tracy) Zhang



## What exactly is linear algebra?

Linear Algebra  $\approx$



1

We would like to compare the Linear Algebra Wikipedia page to the following Wikipedia pages:

1. Philosophy
2. Algebra
3. Tajikistan
4. Geometry
5. Mathematics
6. Lindsay Lohan
7. Musk
8. Linear Regression
9. Analytics

First we need to programmatically scrape these pages.

2

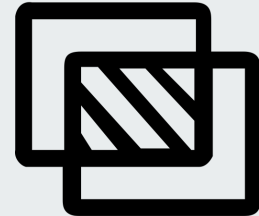
After scraping, we need to clean and vectorize the data, throwing it into a term frequency matrix, with each Wikipedia page corresponding to its own document.

3

Finally, we will need to conduct comparisons amongst the documents using different methods of computing distance and come to a conclusion on which of the nine Wikipedia pages are most similar to the page on “Linear Algebra”.

# Project Objectives

1. Understand the pros & cons of the different methods used for comparing document similarities:
  - a. *Cosine Similarity on Bag of Words Matrix*
  - b. *Cosine Similarity on TF-IDF Matrix*
  - c. *Cosine Similarity applying LSA to TF-IDF Matrix*
  - d. *Euclidean Distance on Bag of Words Matrix*
  - e. *Euclidean Distance on TF-IDF Matrix*
  - f. *Euclidean Distance applying LSA to TF-IDF Matrix*
  - g. *Jaccard Similarity*
2. Come to a conclusion on which of the selected Wikipedia pages is most similar to the page on “Linear Algebra”.



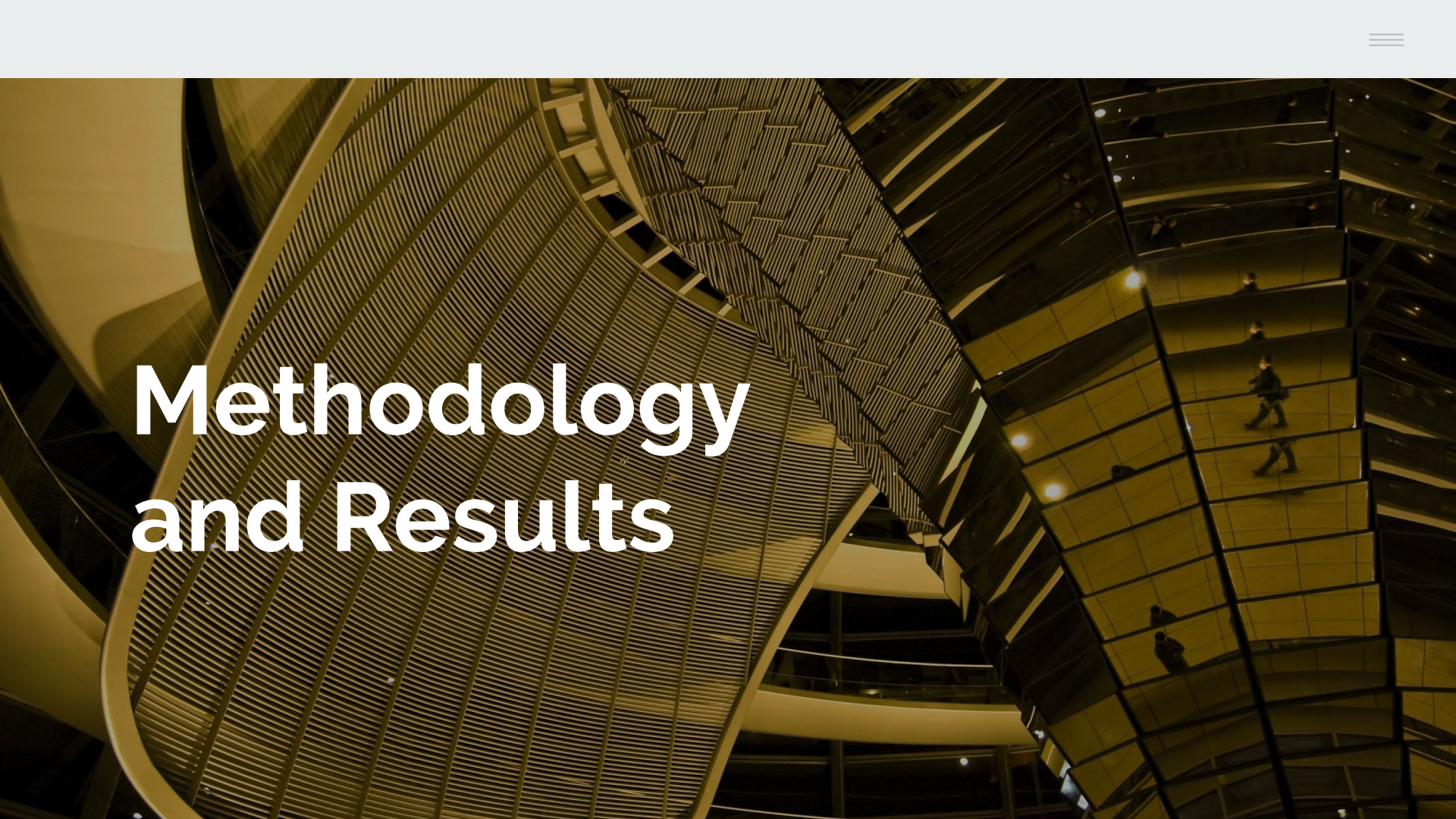


# Role of Linear Algebra

**TF-IDF and Document-Term Matrices:** Transforming words in a series of documents into numerical frequency matrices.

**Document Similarity:** Employing various methods of calculating the distances between vectors within a matrix.

**Latent Semantic Analysis:** Reducing a sparse matrix to a dense matrix of lower rank, thus identifying principle components and eliminating and linearly dependent components.



# Methodology and Results



# Web-Scraping & Data Preparation

1. Utilized Python's requests and BeautifulSoup packages to scrape and parse the html from Wikipedia, respectively.
2. Converted parsed html to a Python string and removed punctuation, stop words, and integers. Also, lemmatized all nouns, verbs, and adjectives.
3. Vectorized cleaned strings and created a combined term frequency matrix made up of **10 documents (rows)** and **4,699 terms (columns)**.



# 3 Ways to Vectorize Documents

1. TF (Term Frequency) creates a **bag of words** that takes into account frequency of words in each document. Since the vectors are not normalized, longer documents bias results.
2. TF-IDF (Term Frequency - Inverse Document Frequency) **measures originality of a word** by comparing the number of times a word appears in a document with the number of documents the word appears in. This means words are weighted such that those that occur frequently in all documents are carry less weight than those that occur in fewer documents.
3. LSA (Latent Semantic Analysis) **finds associations (i.e. linear dependence) between words**. It turns a sparse matrix (e.g. 10x100k) into a dense matrix (e.g. 10x10). If the collection of terms is small, no meaningful associations between words can be derived. LSA requires more processing power.
  - We will use LSA to transform our data first to a **10 document x 10 dimension matrix** and then to a **10 document x 5 dimension matrix**.

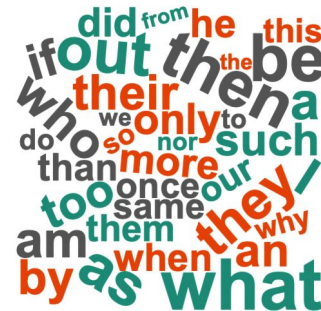
Term Frequencies:											
Sentence	AI	IS	FRIEND	HUMAN	ALWAYS	AND	BEEN	OUR	IT	HAS	
1	1	1	2	0	0	1	1	1	1	1	1
2	1	0	1	1	1	1	1	0	0	0	1

Normalization of Term Frequencies:											
Document	AI	IS	FRIEND	HUMAN	ALWAYS	AND	BEEN	OUR	IT	HAS	
1	0.302	0.302	0.603	0	0	0.302	0.302	0.302	0.302	0.302	0.302
2	0.378	0	0.378	0.378	0.378	0.378	0.378	0	0	0	0.378

$$TF-IDF = TF(t, d) \times IDF(t)$$

Term frequency  $\uparrow$  Inverse document frequency  $\uparrow$   
 Number of times term  $t$  appears in a doc,  $d$   $\log \frac{1 + n}{1 + df(d, t)}$   $\leftarrow$  # of documents  
 Document frequency of the term  $t$



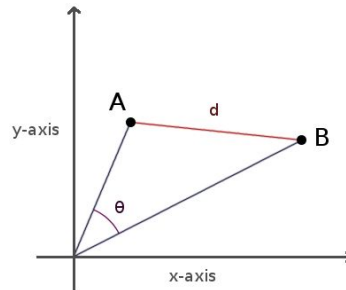
# Euclidean Distance

Euclidean Distance is the one dimensional distance between two points in euclidean space.

This measuring method is common and easy to use. It works well with compact or isolated clusters. However, it is very sensitive to outliers. Therefore, this method is usually not used with sparse or high-dimensionality datasets.

We found that:

1. Linear Algebra has the smallest Euclidean distance from Algebra, Linear Regression, Math, and Geometry using the TF-IDF and LSA vectorization methodologies.
2. The Bag of Words methodology favors Analytics and Musk over Geometry and Math.
3. There is no difference between the TF-IDF distances and the LSA10 distances.
4. The LSA5 distances are all smaller, but directionally consistent with TF-IDF and LSA10.



$$\sqrt{\sum_{i=1}^n (a_i - p_i)^2}$$

Comparison Document	Bag of Words	TF-IDF	LSA10	LSA5
philosophy	281.977	1.385	1.385	1.121
algebra	197.124	1.123	1.123	0.425
tajikistan	268.134	1.403	1.403	1.075
geometry	232.155	1.243	1.243	0.695
math	255.425	1.277	1.277	0.804
lohan	303.962	1.408	1.408	1.199
musk	211.289	1.409	1.409	1.247
linreg	199.379	1.216	1.216	0.636
analytics	223.683	1.394	1.394	1.076



# Cosine Similarity

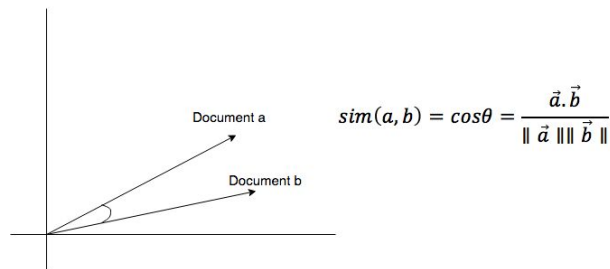
Document similarity can be calculated by measuring the cosine of the angle between two vectors that represent the documents. The **cosine of a small angle is closer to 1** and the **cosine of a large angle is closer to -1**. Cosine Similarity benefits from the fact that it is **not influenced by length of the vector** like Euclidean Distance is.

We found that:

1. Linear Algebra has the smallest angle with Algebra, Linear Regression, Geometry, and Math across all vectorization methodologies.
2. There is no difference between the TF-IDF angles and the LSA10 angles.
3. **For LSA5, the angles between Linear Algebra and related fields were even smaller than those for LSA 10/TF-IDF.**

*Cosine Similarity and Euclidean Distance will give the same directional results with normalized vectors.*

## Cosine Similarity



Comparison Document	Bag of Words	TF-IDF	LSA10	LSA5
philosophy	0.074	0.041	0.041	-0.171
algebra	0.387	0.370	0.370	0.849
tajikistan	0.037	0.016	0.016	-0.207
geometry	0.254	0.228	0.228	0.570
math	0.200	0.185	0.185	0.474
lohan	0.020	0.008	0.008	0.074
musk	0.019	0.008	0.008	0.041
linreg	0.357	0.261	0.261	0.689
analytics	0.051	0.029	0.029	0.069

# Jaccard Similarity

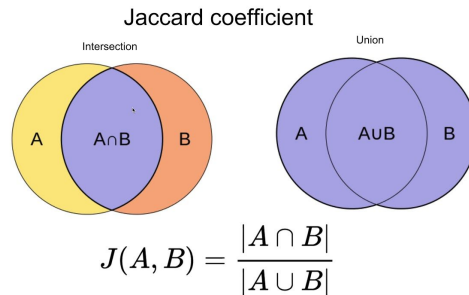
Jaccard Similarity is the **simplest way** of comparing document similarities. It divides the intersection of two documents by their union. The output Jaccard Similarity Coefficient can range from 0 to 1, and **higher coefficients represent higher similarity**.

Despite its simplicity, this method has **some fundamental issues**:

- **Length of document has a huge impact and Weight of words is irrelevant.** Words that appear in many documents are weighted the same as those that appear in few. Therefore, there is bias towards non-descriptive words and longer documents.

We found that:

1. Algebra is the most similar to Linear Algebra.
2. Unlike Euclidean distance and cosine similarity, Jaccard shows second highest similarity between Linear Algebra and Geometry.



Comparison Document	Jaccard Similarity Score
philosophy	0.106
algebra	0.218
tajikistan	0.074
geometry	0.184
math	0.173
lohan	0.057
musk	0.062
linear regression	0.162
analytics	0.119

# Conclusions

1. We found that the 3 pages that are most similar to Linear Algebra, as calculated by all three of our similarity computation methodologies, are **“Algebra”, “Linear Regression”, and “Geometry”**.
2. Calculating the Euclidean Distances using the Bag of Words vectorization methodology yielded results that were the **most inconsistent** with the other methodologies.
3. Calculating the Cosine Similarity by transforming the TF-IDF sparse matrix into a 10x5 dense matrix via LSA5 yield results that **most clearly differentiated topics** that were similar to Linear Algebra from those that were different.

## Conclusions (Cont.)

1. There is **no difference** in distances or angles computed using the **TF-IDF methodology** and the **LSA 10 methodology** because the 10x4699 TF-IDF matrix and the 10x10 LSA 10 matrix are both Rank 10 (i.e. 10 linearly independent columns/features) and the LSA 10 matrix maintained the same geometric shape as the original TF-IDF matrix while eliminating any linearly dependent (i.e. redundant) dimensions.
2. We believe the reason the Jaccard coefficient indicates that Geometry is more related to Linear Algebra than Linear Regression is because the document for “Geometry” is longer than “Linear Regression”, and Jaccard is known to be biased towards longer documents.



# Thank you.

