Data Wrangling and Vectorization

Lindsey Chenault

MSDS 453: Natural Language Processing

October 5, 2025

Introduction and Problem Statement

This exercise changes unstructured movie reviews into structured data and builds the base for natural language analysis. The dataset has 200 reviews, rapsedand each is a separate "document," with student initials and movie name. Additionally, each document has over 500 words. The review text is the immediate input for vectorization, though the metadata, including genre, review type, and document ID, provide context. In part one, students choose terms for a shared class vocabulary, which leaves the approach qualitative. Applicable terms balance being important in individual reviews and appearing across multiple documents, stressing the feature selection trade-offs—part 2 moves to quantitative analysis. TF-IDF quantified frequency-weighted importance, and Word2Vec developed token-level embeddings (Mikolov et al., 2013; Brownlee, 2019). Doc2Vec captured the prevailing meaning of each document, while ELMo produced context-aware embeddings that altered based on surrounding words. These methods build on neural network foundations, as noted by Jurafsky and Martin (2019). Changing the size of the embeddings helped show how extra complexity influenced the quality of the representations**.** There are a few questions to ask during this analysis: Do the terms chosen during the initial review phase also appear among the highest-ranking words in the TF-IDF results across the corpus? How does the number of embedding dimensions affect the way similarity patterns take shape? And do contextual models like ELMo create clearer, more significant clusters than models with static embeddings? Investigating these questions helps refine a shared vocabulary and prepares the dataset for tasks like clustering and classification.

Research Design and Modeling Methods

To prepare the text for analysis, each document was cleaned using standard preprocessing steps: lowercasing, removing punctuation and digits, tokenizing, lemmatizing, filtering out stopwords, and applying stemming. These steps helped cut down on noise and ensured the models received consistent input (Jurafsky & Martin, 2019). TF-IDF focused on unigram tokens and used inverse document frequency to highlight terms that stood out across the corpus. The output included ranked term lists and cosine similarity heatmaps. To go deeper into meaning, Word2Vec was used to explore how words relate to one another semantically. Embedding dimensions of 100, 200, and 300 were experimented with a window size of three, and token subsets (all words, nouns, verbs, and hand-selected terms) were analogized. Heatmaps and clustermaps showed how embedding size and token choice impacted clustering. Doc2Vec extended the analysis to all documents using the Distributed Memory model with a window size of three. Embeddings of 100, 200, and 300 dimensions were compared to determine whether reviews grouped by film or genre, again evaluated with heatmaps.

Finally, ELMo embeddings presented contextual representations. Because of computational limits, only 20 reviews were processed, with document-level vectors averaged from word embeddings and visualized through cosine similarity heatmaps. The design consciously moved from simple to elaborate: TF-IDF for frequency, Word2Vec for token-level meaning, Doc2Vec for document-level meaning, and ELMo for contextual richness. Varying embedding size accentuated trade-offs between interpretability, semantic depth, and computational cost.

Analysis and Interpretation

Only a subset of results is presented in this paper; additional experiments are documented in Colab.

TF-IDF. Selected terms (annie, helen, wedding, wiig) from the established documents had moderate mean scores, with annie (1.85) having the highest. This result proved important within individual reviews but had limited prevalence across the corpus, which illustrates why some document-specific terms are weak for clustering. By contrast, corpus-wide terms (film, movi, barbi, like, time) scored higher (1.67–3.55) but were essentially generic. The cosine similarity heatmap showed little cross-document similarity, reinforcing TF-IDF's limitations for capturing relationships beyond frequency.

Word2Vec. Across embedding dimensions, Word2Vec seized stable semantic relationships. At 100 dimensions, clusters overlapped; at 200 dimensions, groupings of character names, adjectives, and action terms became clearer; at 300 dimensions, gains diminished. A 200-dimensional heatmap showed strong neighborhoods among common tokens (film, movie, character) and genre-linked words (fight, action, scene). Overall, 200 dimensions offered the best balance between clarity and semantic richness.

Doc2Vec. Document-level embeddings grouped reviews by film and, in some cases, by genre. A 300-dimensional t-SNE scatterplot showed tight, movie-specific clusters, while a 100-dimensional similarity heatmap revealed block-like structures reflecting intra-movie cohesion. Higher dimensions produced sharper separability, which confirms Doc2Vec's strength in preserving document-level meaning compared to word-level models.

ELMo. Contextual embeddings produced the clearest clusters. A t-SNE plot showed compact, movie-specific groupings (e.g., *Mean Girls*, *Batman*), while the heatmap displayed substantial intra-movie similarity with faint genre overlaps. Unlike Doc2Vec, ELMo produced more compact clusters, which emphasizes the advantage of contextualized, pretrained embeddings.

Conclusion

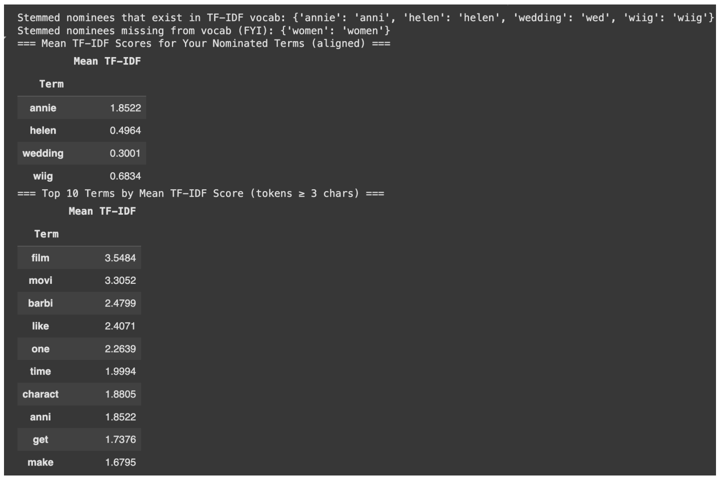
This assignment compared multiple vectorization methods for representing a corpus of movie reviews. TF-IDF quantified distinctive terms but struggled to capture relationships across documents. Word2Vec unveiled unambiguous relationships between words, with 200 dimensions showing a strong middle ground between detail and readability. Doc2Vec took things further by catching the overall meaning of each review, which produced clusters that frequently lined up with the movie or genre, especially when using higher dimensions. ELMo stood out the most, using context-aware, pretrained embeddings to separate reviews more distinctly than the other models.

Overall, the results show the give and take between basic frequency-based methods and more sophisticated embeddings. TF-IDF is straightforward and easy to interpret, but it misses the deeper connections between words and documents. In contrast, models like ELMo do a better job of grasping meaning in context and show the most potential for tasks like clustering and classification.

Results

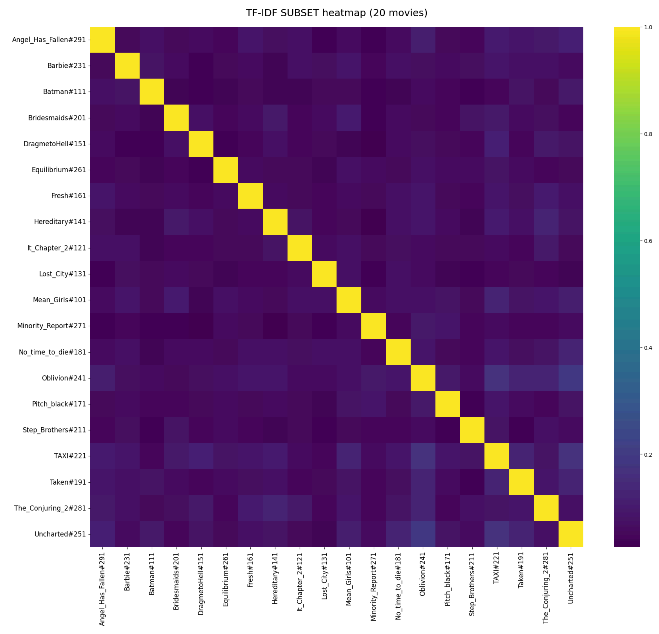
TF-IDF

Mean TF-IDF scores for nominated terms (*Annie, Helen, Wedding, Women, Wiig*)

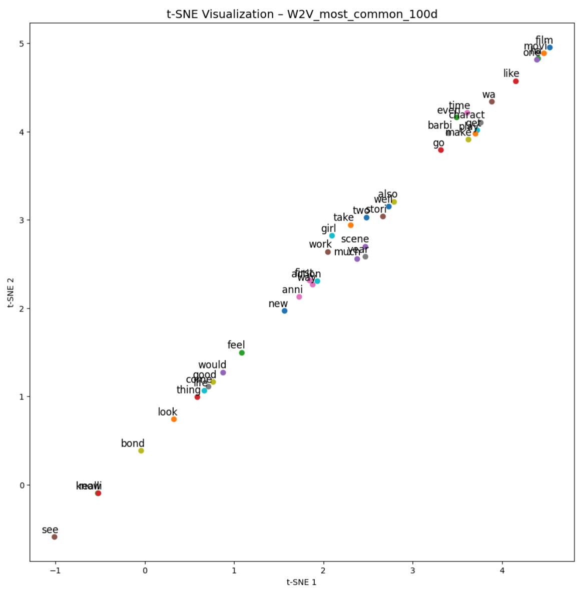


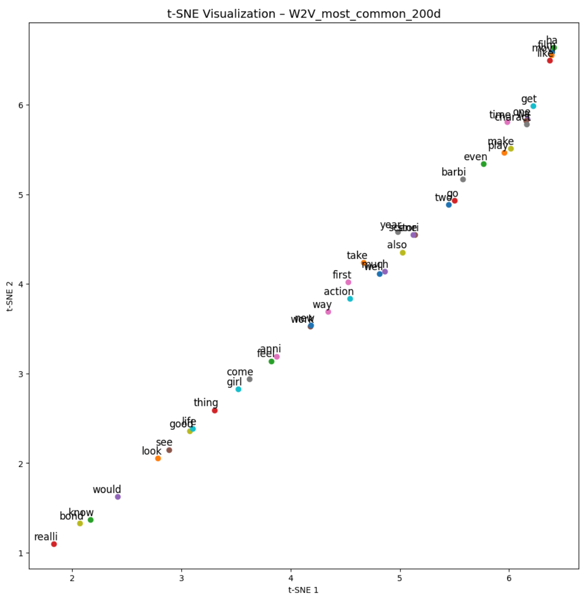
```
Stemmed nominees that exist in TF-IDF vocab: {'annie': 'anni', 'helen': 'helen', 'wedding': 'wed', 'wiig': 'wiig'}
Stemmed nominees missing from vocab (FYI): {'women': 'women'}
=== Mean TF-IDF Scores for Your Nominated Terms (aligned) ===
         Mean TF-IDF
   Term
  annie       1.8522
  helen       0.4964
wedding       0.3001
   wiig       0.6834
=== Top 10 Terms by Mean TF-IDF Score (tokens ≥ 3 chars) ===
         Mean TF-IDF
   Term
   film       3.5484
   movi       3.3052
  barbi       2.4799
   like       2.4071
    one       2.2639
   time       1.9994
charact       1.8805
   anni       1.8522
    get       1.7376
   make       1.6795
```

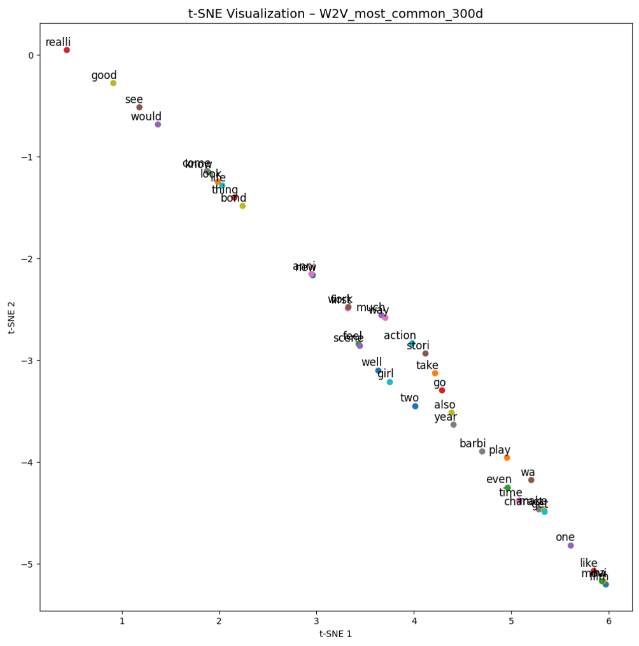
Top 10 highest TF-IDF terms across the corpus

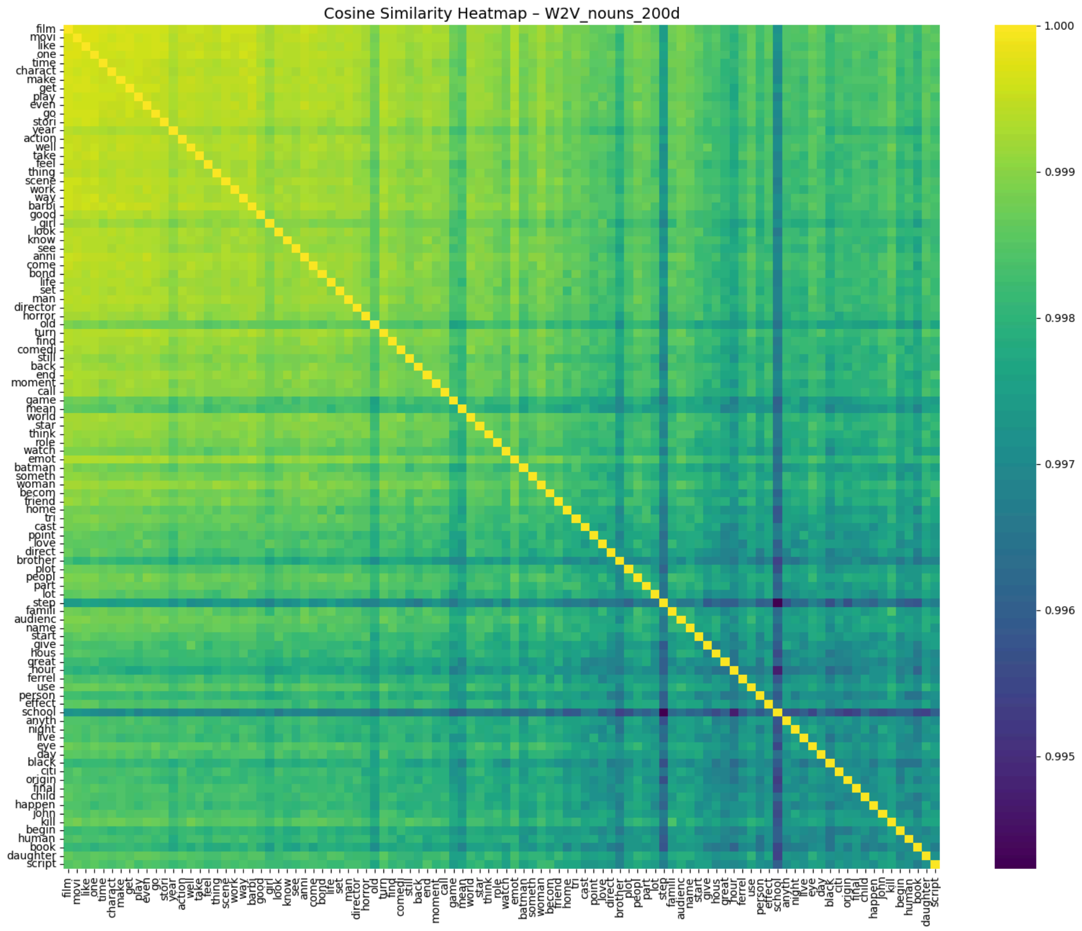


Word2Vec


t-SNE Visualization – W2V_most_common_100d
t-SNE 1
t-SNE 2
film
movi
one
like
wa
time
even
charact
barbi
play
get
make
go
also
well
two
stori
take
girl
scene
work
year
much
first
action
way
anni
new
feel
would
good
come
life
thing
look
bond
knew
realli
see

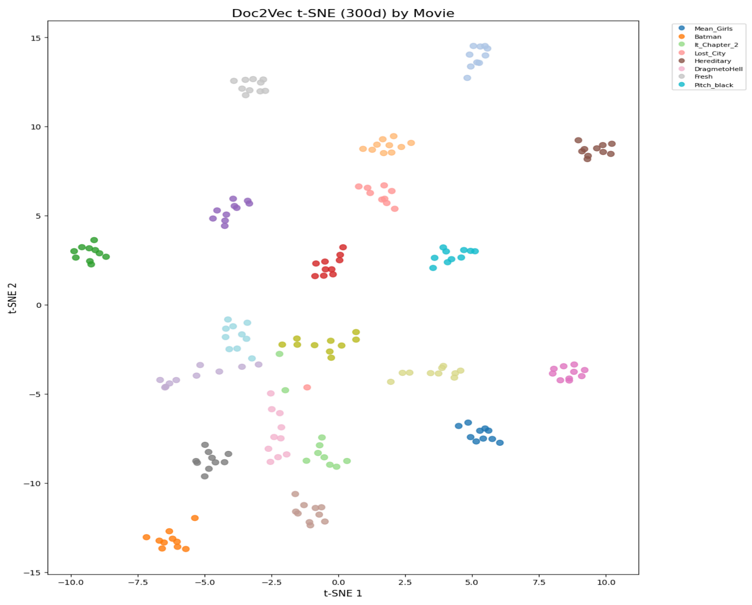

t-SNE Visualization – W2V_most_common_200d
t-SNE 1
t-SNE 2
ha
film
movi
like
get
time
one
charact
make
play
even
barbi
two
go
year
scene
stori
also
take
much
well
first
action
way
new
work
anni
feel
come
girl
thing
life
good
see
look
would
knew
bond
realli


t-SNE Visualization – W2V_most_common_300d
t-SNE 1
t-SNE 2
realli
good
see
would
come
know
look
life
thing
bond
anni
new
work
first
much
way
feel
scene
action
stori
well
girl
take
go
two
also
year
barbi
play
wa
even
time
charact
make
get
one
like
movi
film

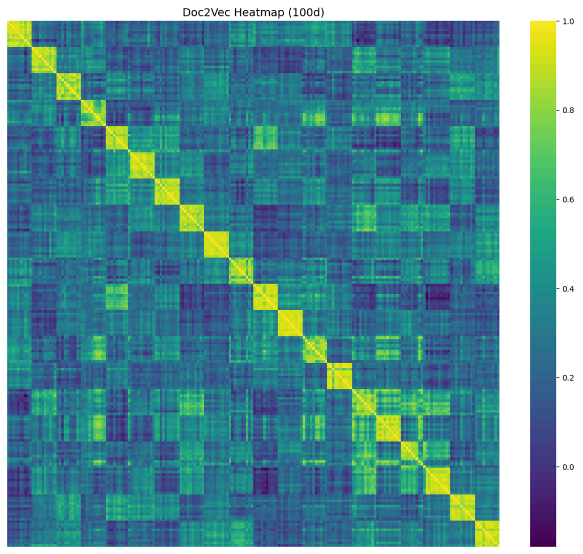

Cosine Similarity Heatmap – W2V_nouns_200d
film
movi
like
one
time
charact
make
get
play
even
go
stori
year
action
well
take
feel
thing
scene
work
way
barbi
good
girl
look
know
see
anni
come
bond
life
set
man
director
horror
old
turn
find
comedi
still
back
end
moment
call
game
mean
world
star
think
role
watch
emot
batman
someth
woman
becom
friend
home
tri
cast
point
love
direct
brother
plot
peopl
part
lot
step
famili
audienc
name
start
give
hous
great
hour
ferrel
use
person
effect
school
anyth
night
live
eye
day
black
citi
origin
final
child
happen
john
kill
begin
human
book
daughter
script
1.000
0.999
0.998
0.997
0.996
0.995

Doc2Vec

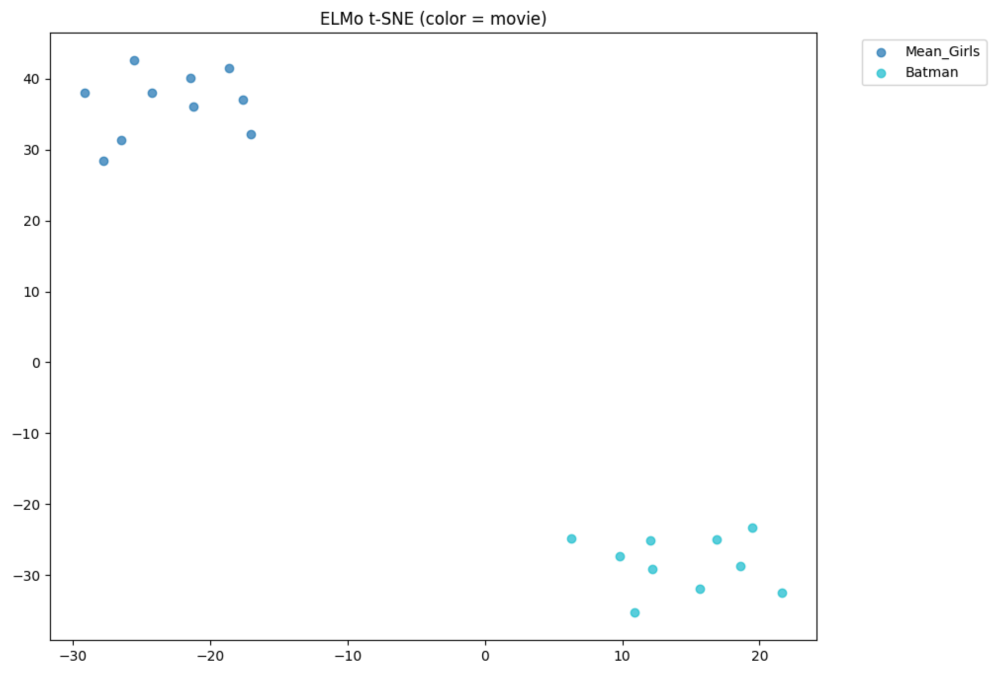
One t-SNE scatterplot (documents separated by movie/genre)



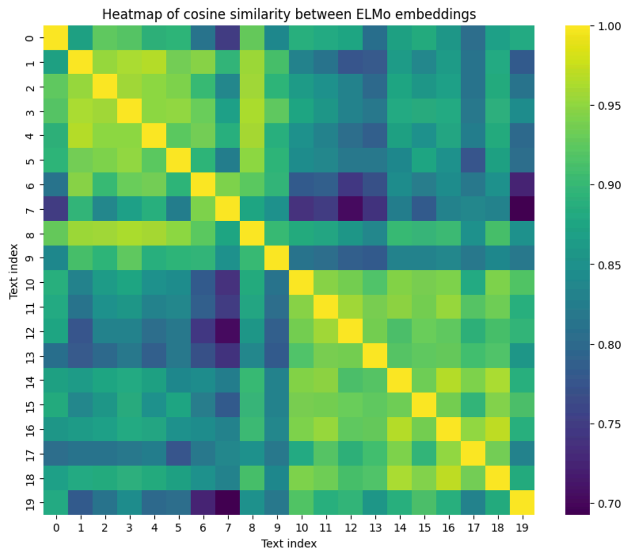
One heatmap or clustermap showing block structures (document clustering).



t-SNE scatterplot colored by movie (shows contextual separation)



Cosine similarity heatmap



References

Brownlee, J. (2019). *Deep learning for natural language processing: Develop deep learning models for natural language in Python*. Machine Learning Mastery.

Jurafsky, D., & Martin, J. H. (2019). *Speech and language processing* (3rd ed., draft). Retrieved from https://web.stanford.edu/~jurafsky/slp3/