

Day 3 - Text (Advanced)

UMN CSS Workshop 2025

Instructor: Alvin Zhou

Learning Goals

- Learn word embedding
- Understand topic modeling and its use in social science
- Explore creative uses of computational text methods
- Practice using external text APIs and topic modeling

Last Class

- Dictionary-Based Methods
- Traditional Classifier (pre-2020)
 - Create features (independent variables) from text: n-grams (bag of words), metadata (emoji, length), TF-IDF, dictionaries (LIWC, MFD)
 - For example, the text “I love data science! 🧪” can have these features
 - "I": 1, "love": 1, "data": 1, "science": 1, “🧪”: 1
 - No. of Characters: 22
 - Punctuation Count: 1
 - "PRP (personal pronoun)": 1, "VBP (verb)": 1, "NN (noun)": 2, "SYM (symbol)": 1
 - The Y (dependent variable) is binary (uncivil/civil; relevant/irrelevant, etc.)
 - Fit different models (e.g., logistic regression with regularization) with training data
 - Evaluate how the model performs
 - Apply the model to the remaining dataset

Limitations of Traditional Features

- Bag of Words and TF-IDF are sparse and high-dimensional
- Vocabulary is rigid: struggles with new/rare words
- Ignores word meaning and semantic similarity
 - “happy” and “joyful” are treated as unrelated
- Doesn’t capture context: “good” \neq “not good”
- Performance plateaus in many classification tasks

Word Embedding

- Each word is represented by a vector
- Words with similar meanings are close in vector space
- Captures semantic and syntactic similarity
 - “king” – “man” + “woman” \approx “queen”
 - “love” and “like” have similar embeddings
- The “GloVe 6B 300-dimensional” model from Stanford
 - trained on Wikipedia + Gigaword, in which:
 - "man" \rightarrow [0.20217, 0.12963, -0.17952, -0.00857, 0.05105, -0.32036, 0.18403, 0.39143, 0.46551, -0.22903, ...]
 - `cosine_similarity("man", "woman") \approx high`

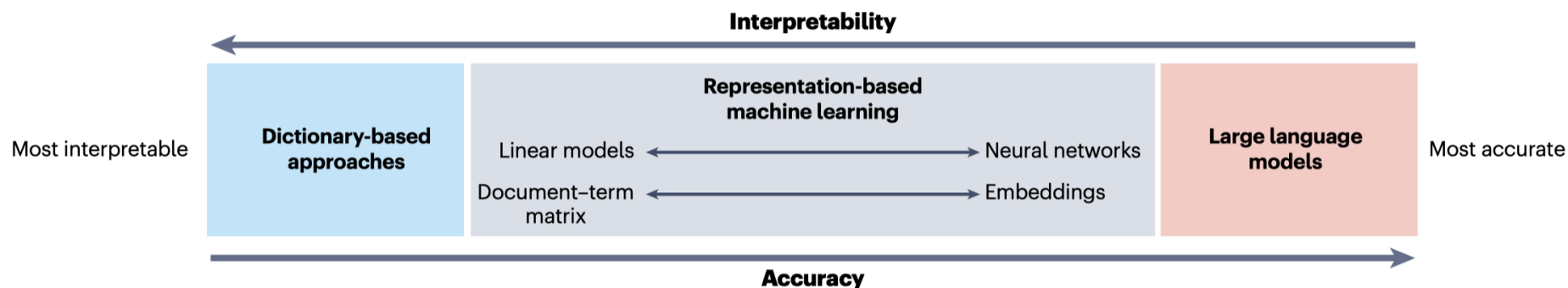
Why Use Word Embeddings? (post-2020)

- Low-dimensional: ~100–300 dimensions vs. 10,000s for BoW
- Generalizes across similar terms (e.g., “happy”, “joyful”)
- Pretrained models available (Word2Vec, GloVe, BERT) – simply download to your computer and you are good to go
- Reduces need for manual feature engineering (e.g., L1/L2 regularization)
- Improved performance in many areas of NLP

How Embeddings Are Used in Practice

- Average word vectors for each document, so that one document is represented by a vector. If
 - “man” is $[1, 0, 1]$
 - “and” is $[1, 0, 0]$
 - “woman” is $[1, 1, 1]$
 - A document with “man and woman” is represented as $[1, 0.333, 0.666]$
- TF-IDF-weighted average
 - If “man” appears too many times in the document, its influence on “averaging word vectors for each document” decreases
- Sentence or document embeddings (e.g., Sentence-BERT)
- **Feed into any classifier (logistic, SVM, NN)**

TF-IDF vs. Word Embedding

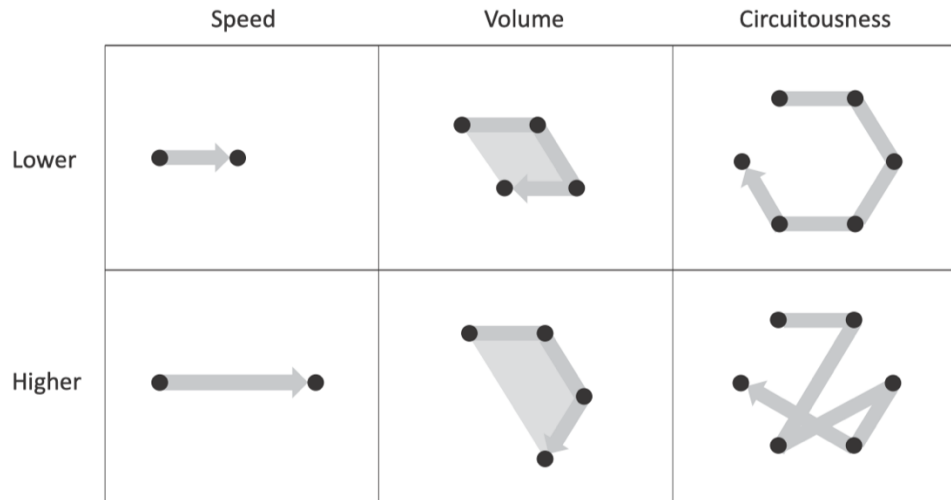


Feature Type	TF-IDF	Word Embedding
Sparse/Dense	Sparse	Dense
Dimensionality	High	Low
Handles Synonyms?	No	Yes
Context-Aware?	No	Sometimes (if BERT)
Pretrained?	Not really	Yes (Word2Vec, GloVe, BERT)
Model Input	Classic ML	ML + DL

Presentation: Toubia et al. (2021)

Presentation: Toubia et al. (2021)

- How quantifying the shape of stories predicts success
- Track emotional arc over time using NLP
- Shape features predict box office revenue
- Social science application: narrative structure & diffusion



You can fine-tune embeddings for your study

- TF-IDF studies usually “start from scratch”
- But studies using embeddings usually grab an existing model and fine-tune it with new data
- “Fine-tuning” means continuing to train a pretrained model on a new, domain-specific corpus, so it adapts to the language and meaning specific to your data.

For Example

- Fine-Tuning GloVe on Social Media
 - Let's say you're studying vaccine discourse on Twitter. You start with GloVe (trained on news/Wikipedia), but you want to adapt it to Twitter slang + COVID-specific terms.
 - In Python (conceptually):

```
from gensim.models import Word2Vec
from gensim.models.keyedvectors import KeyedVectors

# Load pretrained word vectors (e.g., from GloVe)
pretrained_model = KeyedVectors.load_word2vec_format("glove.6B.300d.txt", binary=False)

# Load your own corpus (e.g., tokenized tweets)
custom_corpus = [["get", "vaccinated", "bro"], ["vax", "saves", "lives"]]

# Initialize a Word2Vec model with pretrained weights
model = Word2Vec(vector_size=300, window=5, min_count=1)
model.build_vocab(custom_corpus)
model.build_vocab([list(pretrained_model.key_to_index.keys())], update=True)
model.wv.vectors_lockf = np.ones(len(model.wv), dtype=np.float32) # allows updating
model.wv.intersect_word2vec_format("glove.6B.300d.txt", binary=False, lockf=1.0)

# Fine-tune on your corpus
model.train(custom_corpus, total_examples=len(custom_corpus), epochs=5)
```

For Example

- You get an embedding model that understands “vax,” “jabbed,” “anti-vaxxer” as used in your data, but still retains core semantic structure from GloVe.
- <https://pmc.ncbi.nlm.nih.gov/articles/PMC9578521/>
 - Our study investigated and compared public sentiment related to COVID-19 vaccines expressed on 2 popular social media platforms—Reddit and Twitter—harvested from January 1, 2020, to March 1, 2022.
 - To accomplish this task, we created a fine-tuned DistilRoBERTa model to predict the sentiments of approximately 9.5 million tweets and 70 thousand Reddit comments. To fine-tune our model, our team manually labeled the sentiment of 3600 tweets and then augmented our data set through back-translation. Text sentiment for each social media platform was then classified with our fine-tuned model using Python programming language and the Hugging Face sentiment analysis pipeline.

You can definitely train it from scratch

- Presentation: Kozlowski et al. (2019)

Presentation: Kozlowski et al. (2019)

- Trained custom Word2Vec embeddings on Google Books Ngram data from each decade (1900–1999)
 - Downloaded the Google Books Ngram corpus, stratified by decade (1900–1999).
 - Trained separate Word2Vec models for each decade.
 - Used these decade-specific embeddings to:
 - Measure how meanings of cultural terms shifted across time.
 - Project words into latent dimensions (e.g., class, gender, affluence) defined by antonym word pairs (e.g., rich vs. poor).

Other Text-Based Methods

- **Choose methods to answer your questions!**
 - Named entity recognition
 - PoS Tagging
 - Topic models

Presentation: Knight (2022)

Presentation: Knight (2022)

- The New York Times (1890–1934) and Wall Street Journal (1905–1934)
 - OCR, 400,000+ articles
- Analyzed “agentic talk” — linguistic indicators that describe corporations as intentional actors (e.g., “decided,” “believed”)
- Named entity recognition (NER) to identify organizations
 - Standard Named Entity Recognizer
 - Authors’ own + manual cleaning
- Parsing
 - “nsubj” - the active subject in a sentence
 - Stanford Tagger + manual cleaning
- Dictionary for Verbs
 - Harvard General Inquirer database
 - “cognitive orientation” and “communicating”
- STM

Topic Models

- An unsupervised method to discover hidden thematic structure
 - “what do these documents/text talk about?”
- Useful for exploratory analysis when labels aren't available
- Widely popular in social sciences

Basic Logic of Topic Modeling

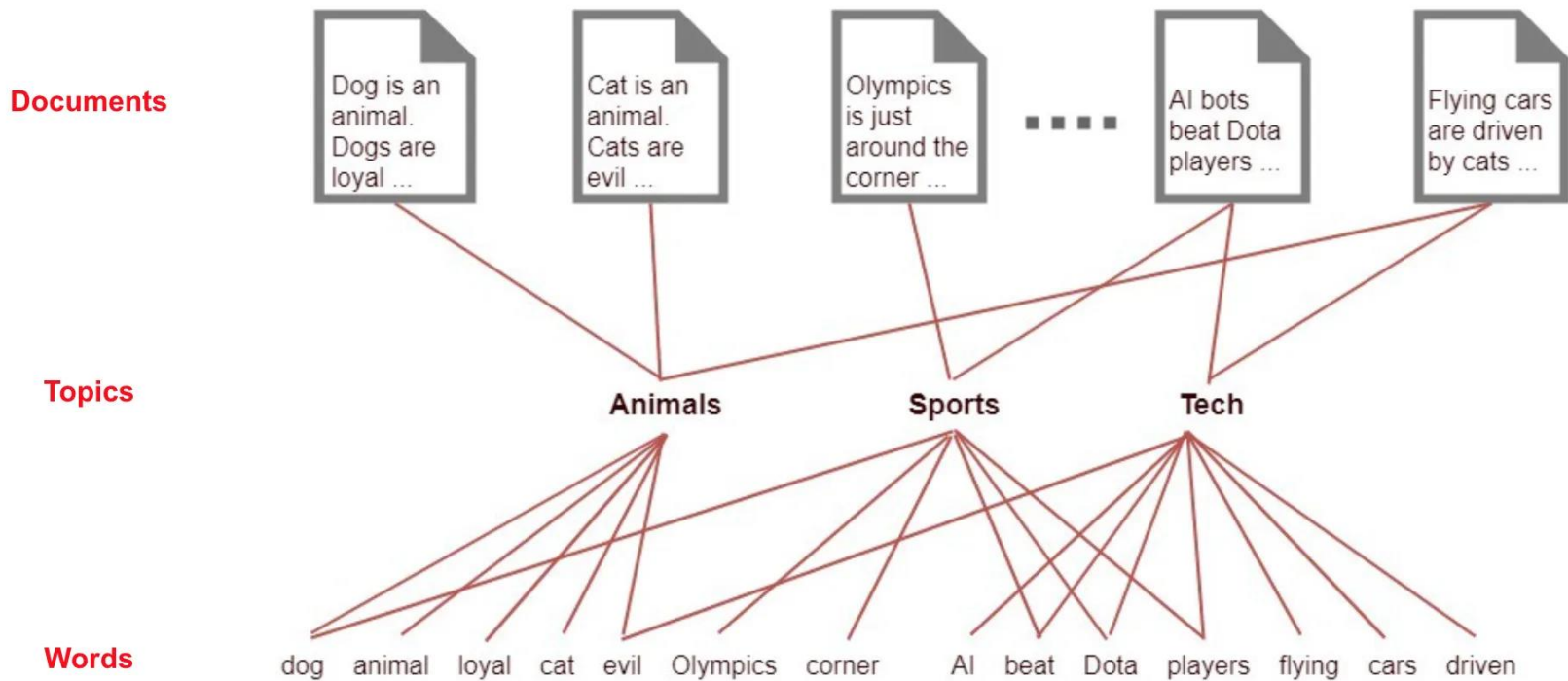
- Three Levels:
 - Documents
 - Topics
 - Words
- Documents are composed of multiple topics, with weights
- Each topic is described by a set of words, with weights

Basic Logic of Topic Modeling

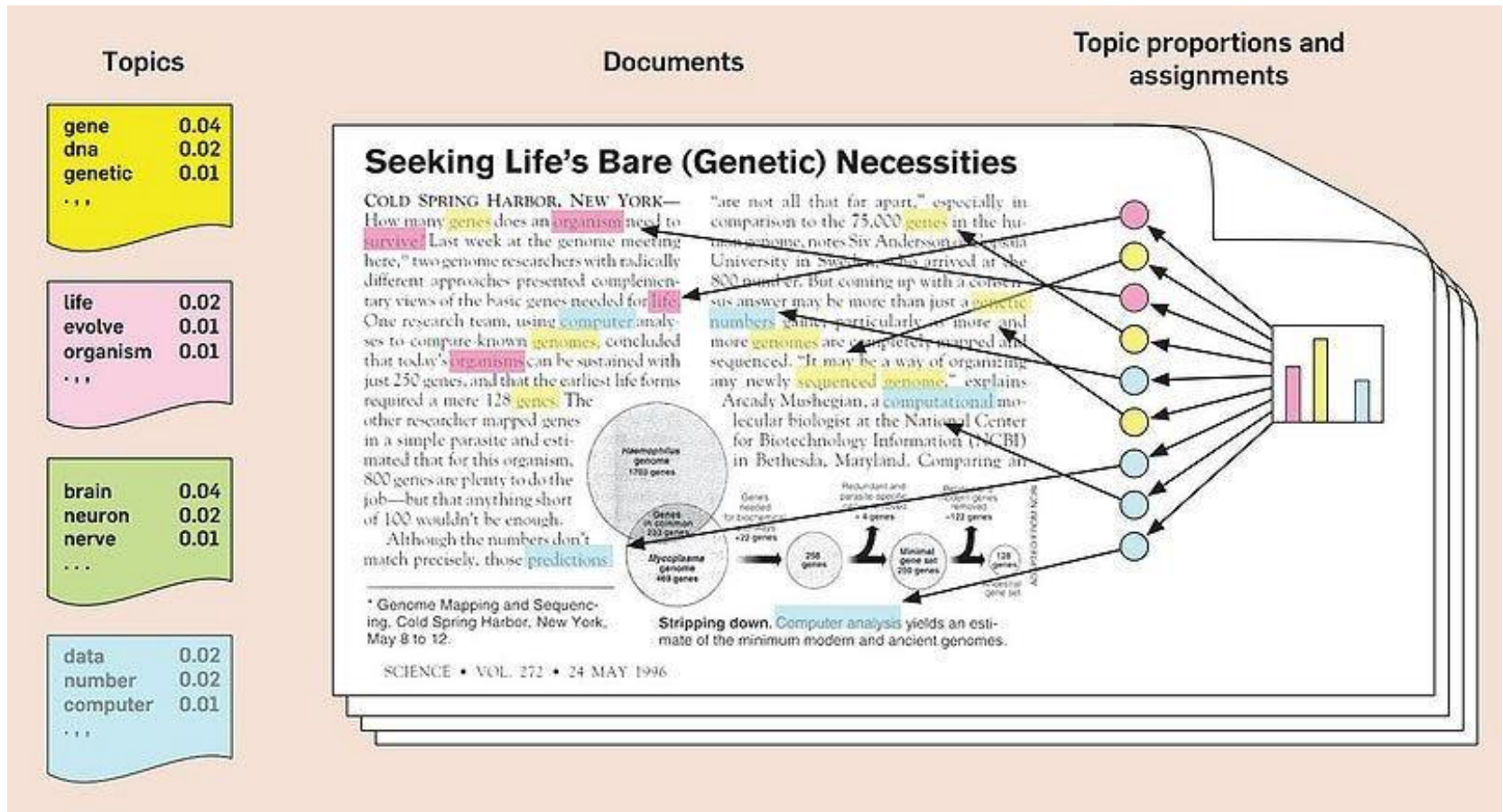
- Early/Most topic models based on TF-IDF
- Newer models based on word embedding
- If two words (word 1 and word 2) always co-occur, the computer understands that they might belong to the same topic

	words1	words2	words3	words4	words5
doc1	0	0	1	0	0
doc2	2	0	1	1	0
doc3	0	0	1	1	0
doc4	0	0	1	1	1

Basic Logic of Topic Modeling



Basic Logic of Topic Modeling



Latent Dirichlet Allocation (LDA)

- Introduced by Blei, Ng, and Jordan (2003)
- Assumption
 - Each document is a mixture of topics
 - Each topic is a mixture of words
 - **Both mixtures are drawn from Dirichlet distributions**
- Widely used, but with limitations:
 - Cannot integrating metadata
 - Doesn't work well with short text

Structural Topic Modeling (STM)

- Extends LDA by incorporating document-level metadata/covariate
 - Year
 - Political Party
 - Before/After Treatment
- Instead of assuming Dirichlet distributions, it uses logistic-normal prior (has a normal distribution in log-space)
 - Don't ask me what they mean 😊
- Allows topic prevalence and content to vary by covariates
- Excellent for theory-testing in social science
- Implemented in R using stm package

LDA/STM Output Example

- You get:
 - A list of topics (top words per topic)
 - Topic 1's top words: animal, cat, dog, cow, farm, ...
 - Topic 2's top words: work, employ, farm, rice...
 - ...
 - Document-topic proportions
 - This document consists of 50% Topic 1, 25% Topic 2, 15% Topic 3, and 10% Topic 4
- Specific to STM:
 - Topic correlations across all documents
 - How likely Topic 1 and Topic 2 can co-occur (i.e., a network)
 - Covariate effect (e.g., topic prevalence by party/time/treatment)

Topic Modeling Does Not Give You Labels

- You get:
 - A list of topics (top words per topic)
 - Topic 1's top words: animal, cat, dog, cow, farm, ...
 - Topic 2's top words: work, employ, farm, rice...
 - ...
- But they never tell you what the topics are, the computer only knows that these words seem to co-occur here
 - No labels like “animal,” “farming,” or “climate change” / “terrorism” etc
 - Labels are interpretive, researcher-assigned summaries
 - Needs researchers' substantial expertise to cover research areas
- You must:
 - Inspect top words
 - Read exemplar documents for each topic
 - Assign a topic label and discuss with co-authors

Topic Modeling Does Not Give You Labels

- Topic modeling is a discovery tool, not a labeling machine.
- It helps uncover patterns — you provide the meaning.
- A topic with top words like “children,” “school,” “teacher,” “lunch,” “bus” might sound like it’s about “Education”
- But since I am aware that the documents I have are all NY Times articles mentioning “gun,” I should know that it’s mostly about “school shootings.”
- If I did the search about “poverty,” then I should label it “childhood hunger.”
- Reading top representative documents is also helpful.

Specify the No. of Topics (K) (LDA/STM)

- There is no “correct” K , as much art as science
 - It’s a modeling choice, not a ground truth
 - Too few topics → overgeneralized, mixed themes
 - Too many topics → fragmented, redundant, hard to interpret
 - Need to find a balanced K with human interpretation
- Model diagnosis does help
 - Held-out likelihood (predictive performance)
 - Semantic coherence (do top words make sense together?)
 - Exclusivity (are top words unique to topics?)
 - But most important: **Interpretability** (read top documents, qualitative)
- Tip
 - Plot “Exclusivity vs. Semantic Coherence” to find a sweet spot
 - Favor interpretability over statistical fit for most social science audiences
 - Be transparent and communicate the transparency (in appendix)

Limitations of LDA/STM

- Bag-of-words assumption: no word order
- Topics can be hard to interpret
- Sensitive to preprocessing choices
- STM assumes linear effects of covariates

What Comes After STM?

- BERTopic: combines transformer embeddings with clustering
 - **No need to specify topic count in advance**
 - More coherent topics via sentence-level context
 - Better for short texts (e.g., tweets)
- Not always good
 - You might get 15 topics, but your theory posits 6 dimensions
 - So when social scientists use BERTopic (which typically returns more topics than they want), they have to do *post-hoc merging*
 - Merge topics 3, 8, and 14 because they all relate to “government surveillance.”
 - This is defensible but might come off weird in eyes of reviewers
- Instead, use STM, run various K models, and pick the consensus

Topic Modeling Summary

- Use LDA to explore topics of long documents
- Use STM when you have metadata and theory-driven questions
- Use BERTopic for modern, contextual embeddings + better clustering
- All topic models are tools — interpret with care

Case: Zhou et al. (2023)

- Exploring PR research topics via topic modeling
- Use STM to identify topics and clusters
- Compare topic distributions across journals and time
- Network simulation to test inter-cluster dynamics

RQ & Findings

- What topics do public relations scholars study?
- What clusters/themes emerge?
- Do these clusters/themes intersect with each other?

RQ & Findings

- What topics do public relations scholars study?
 - What clusters/themes emerge?
 - Do these clusters/themes intersect with each other?
-
- Identify 65 topics
 - These 65 topics cluster into 9 subfields
 - These subfields do not talk to each other

Data and Methods

- Web Scraping Data
 - Time: from 2010 to 2020
 - Journals:
 - Public Relations Review (PRR)
 - Journal of Public Relations Research (JPRR)
 - 1093 papers from PRR and 200 from JPRR
 - 7,400,685 words

Data and Methods

- Method 1: Structural Topic Modeling
 - We identified 65 topics, such as
 - “Twitter” “Facebook”
 - “Relationship management” “Nonprofit Management”
 - “Image Repair” “Situational Crisis Communication Theory”

Data and Methods

- Method 1: Structural Topic Modeling
 - We identified 65 topics, such as
 - “Twitter” “Facebook” --- “Digital Media”
 - “Relationship management” “Nonprofit Management” --- “Strategic Management”
 - “Image Repair” “Situational Crisis Communication Theory” --- “Crisis Comm”

Data and Methods

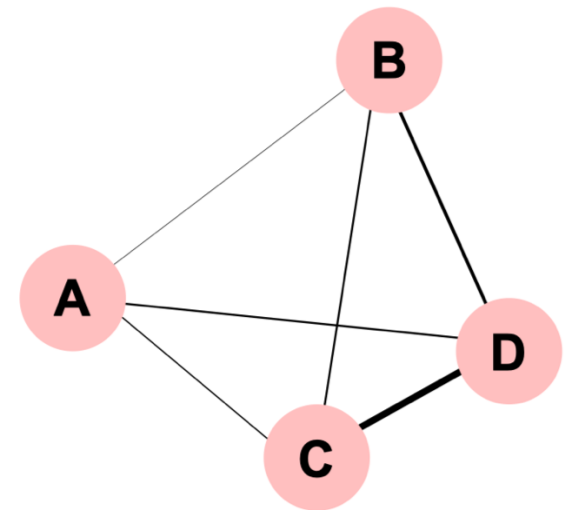
- Method 1: Structural Topic Modeling
 - We identified 65 topics, such as
 - “Twitter” “Facebook” --- “Digital Media”
 - “Relationship management” “Nonprofit Management” --- “Strategic Management”
 - “Image Repair” “Situational Crisis Communication Theory” --- “Crisis Comm”
 - <Like, comment, and share on Facebook: How each behavior differs from the other> is detected to have:
 - Digital Media (73.7%)
 - Strategic Management (18.7%)
 - Public Relations Professionalism (0.1%), Crisis Communication (2.3%), Internal Communication (0.8%), Global Public Relations (0.0%), Rhetoric and Philosophy (0.1%), Media Relations (0.2%), Critical Studies (0.0%)

Data and Methods

- Method 2: Inter-Cluster Network Analysis

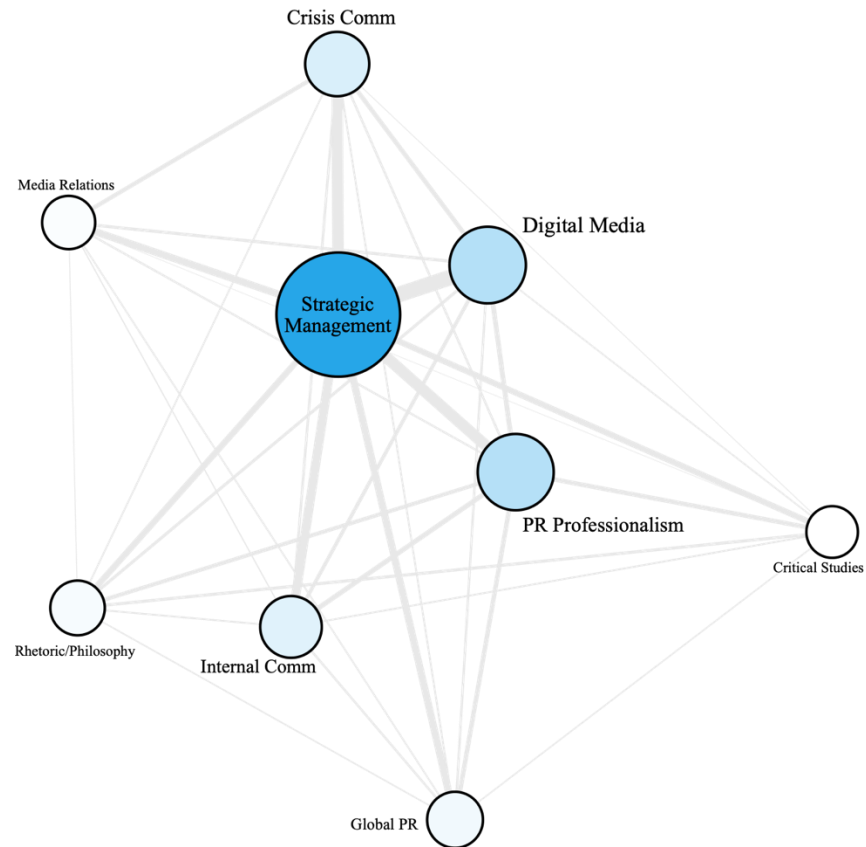
	Cluster A	Cluster B	Cluster C	Cluster D
Article 1	0.1	0.2	0.3	0.4

Article 1's Contribution to the Tie Strength in the Inter-Cluster Network				
	Cluster A	Cluster B	Cluster C	Cluster D
Cluster A	-	$0.1*0.2$	$0.1*0.3$	$0.1*0.4$
Cluster B	-	-	$0.2*0.3$	$0.2*0.4$
Cluster C	-	-	-	$0.3*0.4$
Cluster D	-	-	-	-



Data and Methods

- Method 2: Inter-Cluster Network Analysis



Data and Methods

- Method 3: Network Simulation

Article	Crisis	Digital	Global	...	Management	Critical	Media
1	0.000	0.003	0.001	...	0.749	0.001	0.000
2	0.006	0.002	0.070	...	0.420	0.005	0.003
3	0.204	0.010	0.004	...	0.226	0.025	0.370
4	0.167	0.028	0.031	...	0.124	0.000	0.012
...
1291	0.008	0.061	0.012	...	0.076	0.063	0.007
1292	0.786	0.001	0.002	...	0.102	0.001	0.001
1293	0.003	0.233	0.001	...	0.142	0.082	0.006

Data and Methods

- Method 3: Network Simulation
 - For each article, cluster proportions add up to 100%
 - For each cluster, its proportion across all articles remains the same

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	...	Cluster 7	Cluster 8	Cluster 9
Paper 1								
Paper 2								
Paper 3								
...								
Paper 1292								
Paper 1293								

Data and Methods

- Method 3: Network Simulation
 - For each article, cluster proportions add up to 100%
 - For each cluster, its proportion across all articles remains the same

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	...	Cluster 7	Cluster 8	Cluster 9
Paper 1								
Paper 2								
Paper 3								
...								
Paper 1292								
Paper 1293								

Data and Methods

- Method 3: Network Simulation
 - For each article, cluster proportions add up to 100%
 - For each cluster, its proportion across all articles remains the same

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	...	Cluster 7	Cluster 8	Cluster 9
Paper 1								
Paper 2								
Paper 3			$[i_1, j_1]$			$[i_1, j_2]$		
...								
Paper 1292			$[i_2, j_1]$			$[i_2, j_2]$		
Paper 1293								

Data and Methods

- Method 3: Network Simulation
 - For each article, cluster proportions add up to 100%
 - For each cluster, its proportion across all articles remains the same

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	...	Cluster 7	Cluster 8	Cluster 9
Paper 1								
Paper 2								
Paper 3			$[i_1, j_1] - \Delta$			$[i_1, j_2] + \Delta$		
...								
Paper 1292			$[i_2, j_1] + \Delta$			$[i_2, j_2] - \Delta$		
Paper 1293								

Data and Methods

- Method 3: Network Simulation
 - For each article, cluster proportions add up to 100%
 - For each cluster, its proportion across all articles remains the same
- 1000 Simulated Networks / Alternative Universes/Timelines
- 95%/5% upper/lower bound as the confidence interval for tie strengths
- We simulated “what the field’s interconnection could have been”.

Data and Methods

- Method 3: Network Simulation

[illegible]

Lab Preview

- Explore semantic similarity using word embedding
- Practice Google Perspective API
- Run STM with presidential speech corpus and metadata
- Pick K , generate the model, label topics
- Explore covariates' effects on topical proportion
- Visualize topic correlations
- Download and Install *Gephi*
 - <https://gephi.org>
 - We will use it next class