STEVENS
INSTITUTE OF TECHNOLOGY
1870

Natural Language Processing and Word Embedding

FA690 Machine Learning in Finance

Dr. Zonghao Yang

2025 Spring

Learning Objectives

- Understand text representation techniques: from Bag-of-Words to advanced word embeddings, including Word2Vec and GloVe algorithms
- Identify the limitations of traditional text analysis methods and their solutions
- Apply NLP techniques to analyze and measure corporate culture in financial contexts

Textual Analysis

Text

- Daily: Social media, messages, textbooks, web content
- Finance: 10-K reports, transcripts of earning calls, news articles
- Business: Contracts, patent descriptions, resumes

Features and Challenges

- Raw text consists of an ordered sequence of language elements
  - Length of a sentence varies
  - An ordered sequence
  - Context
- Challenge for analyzing text: High-dimensionality
  - The unique representation of a thirty-word *X* message, using one thousand common English words, has dimension 1000^{30}

Example
- "Alice loves Bob." vs. "Bob loves Alice."
- "My favorite fruit is an **Apple**" vs. "The star product of **Apple** is iPhone."

Why Today?

- Data: An ever-increasing share of human interaction, communication, and culture is recorded as digital text
- Methodology: Natural language processing and understanding
  - Semantic meaning: *Word2Vec*
  - Length and sequence: *Recurrent Neural Networks* (RNN)
  - Context: *Transformer* architecture based on the attention mechanism
  - Large language models (LLMs) are a black-box gigantic neural network that does all above
- Technology: Computing power and resources

- Today: ChatGPT, DeepSeek, Llama, etc.

Textual Analysis Pipeline

Gentzkow, M., Kelly, B., & Taddy, M. (2019). Text as data. *Journal of Economic Literature*, 57(3), 535-574.

- Represent raw text as a numerical array
  - Bag-of-words representation
  - Word2Vec and Glove
  - Embeddings from deep neural networks (e.g., BERT)
- Frame the textual analysis as a prediction problem
  - Take the numerical arrays as the input
  - Train a model to make predictions
- Integrate the text-based predictions with economic models to interpret the economic implications

Example (Bybee et al., 2024, Journal of Finance)
- Use bag-of-words to represent business-related news articles
- Apply topic modelling to estimate the underlying topics of the article
- Build a regression model to predict the macro conditions using the topic distributions

Examples of Text-Related Predictions

- Sentiment analysis
  - What is the sentiment of a review/analyst article/news article?
  - How does business news predict the macroeconomic conditions?
- Consumer confidence: Predict consumer confidence or spending patterns based on product reviews or social media posts
- Company performance: Predict a company's financial performance (e.g., revenue, profit) based on the annual reports, earnings calls, or news
- Credit risk: Predict the creditworthiness of individuals or businesses based on textual data from loan applications, financial statements, or news

Bag-of-Words Representation

- Build the vocabulary
  - Strip out elements like punctuation and numbers
  - Remove stop words: e.g., “and”, “the”, “we”
  - Stemming: e.g., “economic”, “economics”, “economically” => “economic”
- Create the bag-of-words (BOW) representation by counting the occurrence of words for the text body

Example of BoW Representation

- Corpus
  - D1: How time flies! It was the best of times.
  - D2: How time flies! It was the worst of times.
  - D3: It was the age of wisdom.
  - D4: It was the age of foolishness.
- Step 1: Build Vocabulary
- Step 2: Count the number of occurrences
- This word count table is also referred to as term-document matrix

	Best	Worst	Time	Age	Wisdom	Foolish	Fly
D1	1	0	2	0	0	0	1
D2	0	1	2	0	0	0	1
D3	0	0	0	1	1	0	0
D4	0	0	0	1	0	1	0

BoW Representation

TF-IDF Score

$$\text{TF} - \text{IDF Score} = tf_{ij} \times idf_j$$

- i – document, j – word
- Term frequency (tf_{ij}): The count of occurrences of j in i
- Inverse document frequency (idf_j): the log of one over the share of documents containing j, i.e., $\log(n/d_j)$
  - Number of documents containing word j, $d_j = \sum_i 1_{[tf_{ij}>0]}$
  - Intuition: Word j is discounted if it appears in many documents, such as stop words (e.g., 'and', 'the', 'we')
- By excluding the words with tf-idf scores below some cutoff, this approach excludes both common and rare words

BoW Representation with TF-IDS Scores

- Corpus
  - D1: How time flies! It was the best of times.
  - D2: How time flies! It was the worst of times.
  - D3: It was the age of wisdom.
  - D4: It was the age of foolishness.
- Step 1: Build Vocabulary
- Step 2: Calculate tf-idf score for each word in each document

$$\text{TF} - \text{IDF Score} = tf_{ij} \times idf_j = tf_{ij} \times \log(n/d_j)$$

	Best	Worst	Time	Age	Wisdom	Foolish	Fly
D1			2 * log(4/2)				
D2			2 * log(4/2)				
D3			0 * log(4/2)				
D4			0 * log(4/2)				

BoW Representation

Limitations of BoW Representation

- Loss of Word Order and Context: BoW ignores the sequence and structure of words, losing syntactic and contextual information
- Sparsity: BoW creates high-dimensional, sparse vectors, leading to inefficiency in storage and computation (Vector dimension = number of words in vocabulary, e.g., 500,000+)
- Semantic Ignorance: BoW fails to capture word meanings, treating synonyms differently and polysemous words identically
- No Handling of Out-of-Vocabulary Words: BoW cannot represent words not seen during vocabulary creation

- Alternatives? Word embeddings

Semantic Ignorance

- Example: In web search, if a user searches for “New Jersey motel”, we would like to match documents containing “New Jersey hotel”
- Semantic ignorance with BoW representation
  - “motel” = [0 0 0 0 0 1 0 0 0 0 0 0 0 0]
  - “hotel” = [0 0 0 0 0 0 0 1 0 0 0 0 0 0]
  - Issues: These two vectors are orthogonal; there is no natural notion of similarity
- Solution: Learn to encode similarity in the vectors themselves

Word Embedding

Machine learns the meaning of words from reading a lot of documents without supervision, or self-supervision

One-hot Encoding

apple = [1 0 0 0 0]

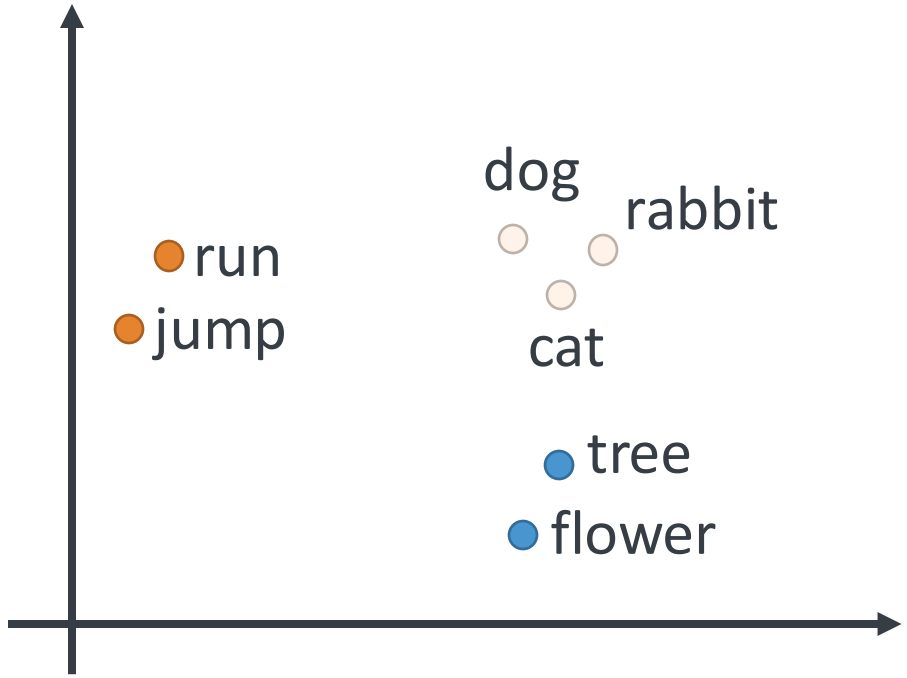
bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

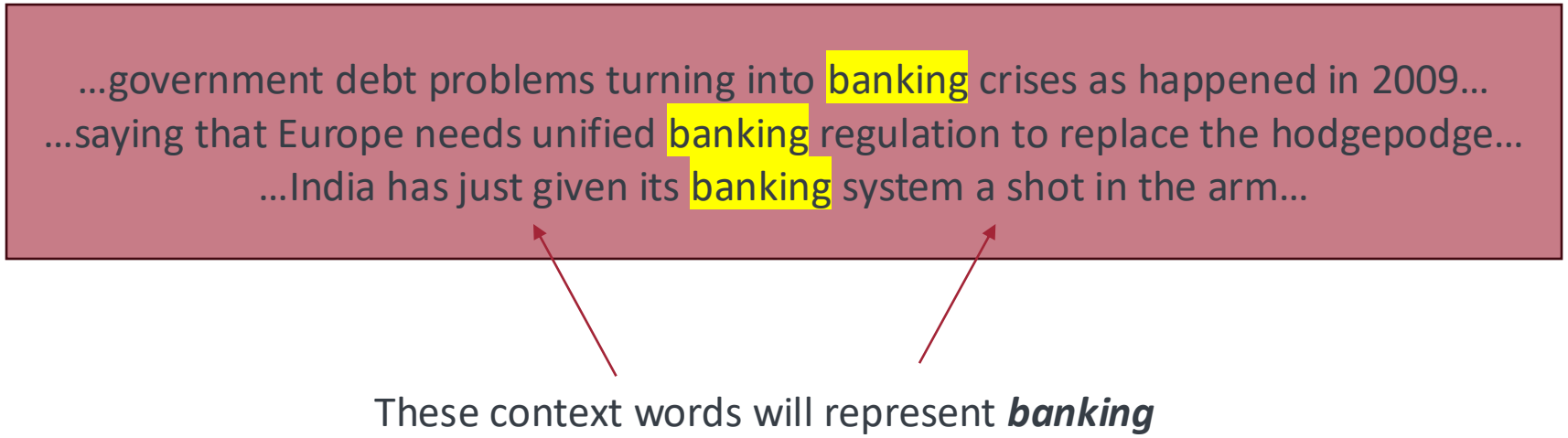
elephant = [0 0 0 0 1]

Word Embedding



Representing Words by Their Context

- Distributional semantics: A word's meaning is given by the words that frequently appear close-by
  - One of the most successful ideas of modern statistical NLP
- When a word w appears in a text, its context is the set of words that appear nearby (within a fixed-size window)
- We use the many contexts of w to build up a representation of w



Word Vectors

We will build a dense vector for each word, chosen so that it is similar to vectors of words that appear in similar contexts, measuring similarity as the vector dot (scalar) product

$$banking = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

$$monetary = \begin{pmatrix} 0.413 \\ 0.582 \\ -0.007 \\ 0.247 \\ 0.216 \\ -0.718 \\ 0.147 \\ 0.051 \end{pmatrix}$$

Note: word vectors are also called (word) embeddings

Word2Vec

Word2Vec

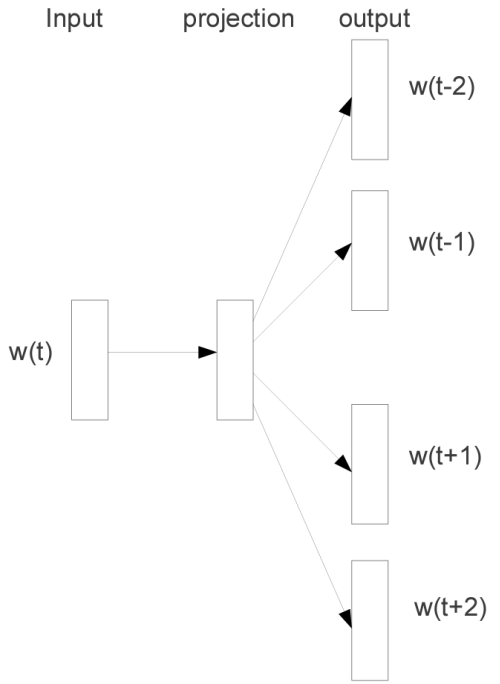
Distributed representations of words and phrases and their compositionality — 46748 — 2013
T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean
Neural information processing systems

Efficient estimation of word representations in vector space — 46051 — 2013
T Mikolov, K Chen, G Corrado, J Dean
arXiv preprint arXiv:1301.3781

NeurIPS 2023 Test-of-Time Award

Word2vec is a framework for learning word vectors (Mikolov et al. 2013)

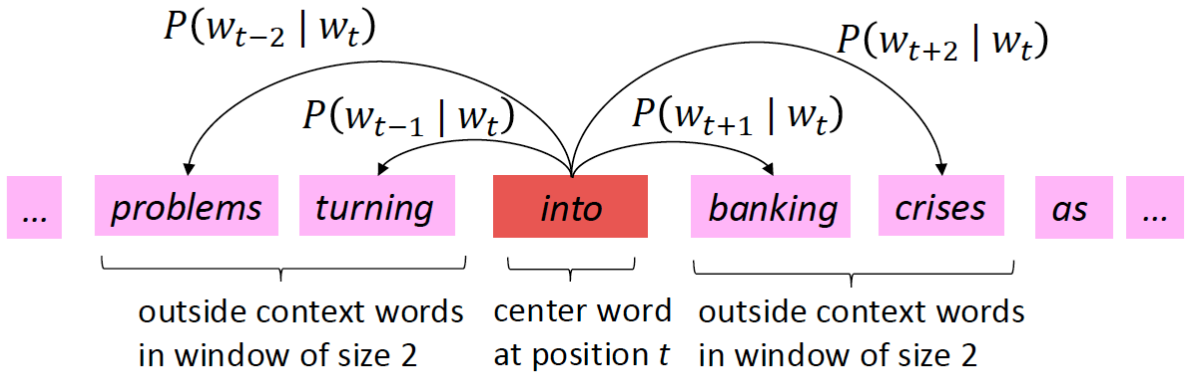
- We have a large corpus of text: a long list of words
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word c and context words o
- Use the similarity of the word vectors for c and o to calculate
  - the probability of o given c (Skip-grams)
  - the probability of c given o (Continuous Bag of Words)
- Keep adjusting the word vectors to maximize this probability

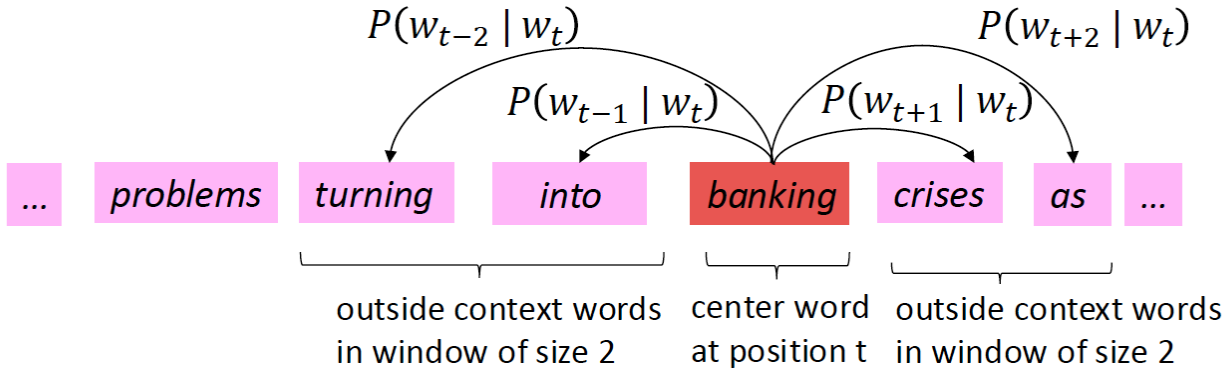


Mikolov et al. (2013): Skip-grams

Word2Vec

Example windows and process for computing $P(w_{t+j}|w_t)$





Word2Vec: Objective Function

- For each position $t = 1, \ldots, T$, predict context words within a window of fixed size m, given center words w_t. Data likelihood:

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \neq 0}} P(w_{t+j} | w_t, \theta)$$

- The objective function $J(\theta)$ is the average negative log likelihood:

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} \log P(w_{t+j} | w_t, \theta)$$

- Minimizing objective function is equivalent to maximizing predictive accuracy

Conditional Probability $P(w_{t+j}|w_t, \theta)$

- How to calculate $P(w_{t+j}|w_t, \theta)$?
- We will use two vectors per word w
  - v_w when w is a center word
  - u_w when w is a context word
- Then for a center word c and a context word o, we use the softmax function to specify the conditional probability

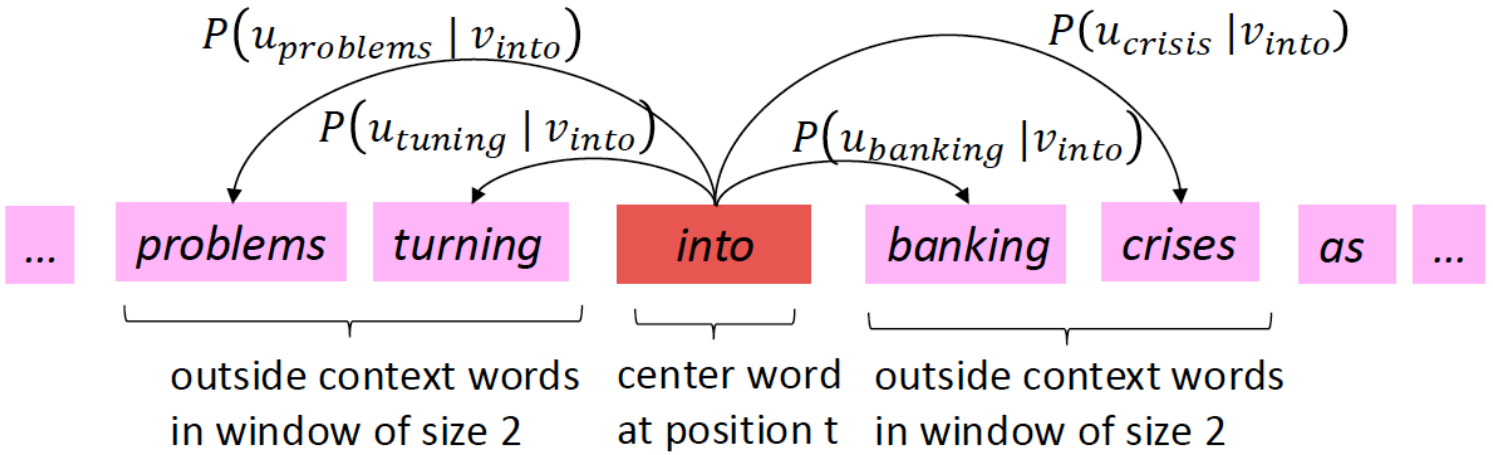
$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

- The objective function

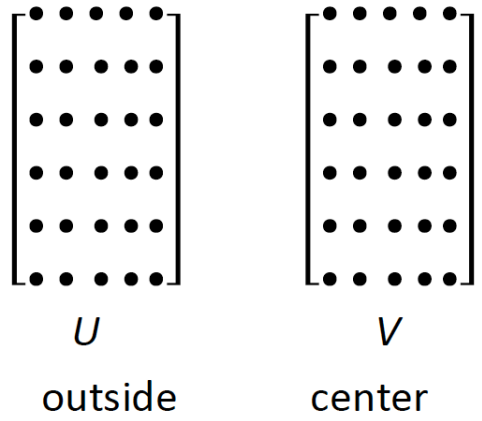
$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} \log P\left(w_{t+j}|w_t, \theta\right) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} \log \frac{\exp(u_{t+j}^T v_t)}{\sum_{w \in V} \exp(u_w^T v_t)}$$

Word2Vec with Vectors

- Example windows and process for computing $P(w_{t+j}|w_t)$
- $P(u_{problems}|v_{into})$ is short for $P(problems|into; u_{problems}, v_{into}, \theta)$



Word2Vec: Skip-Grams

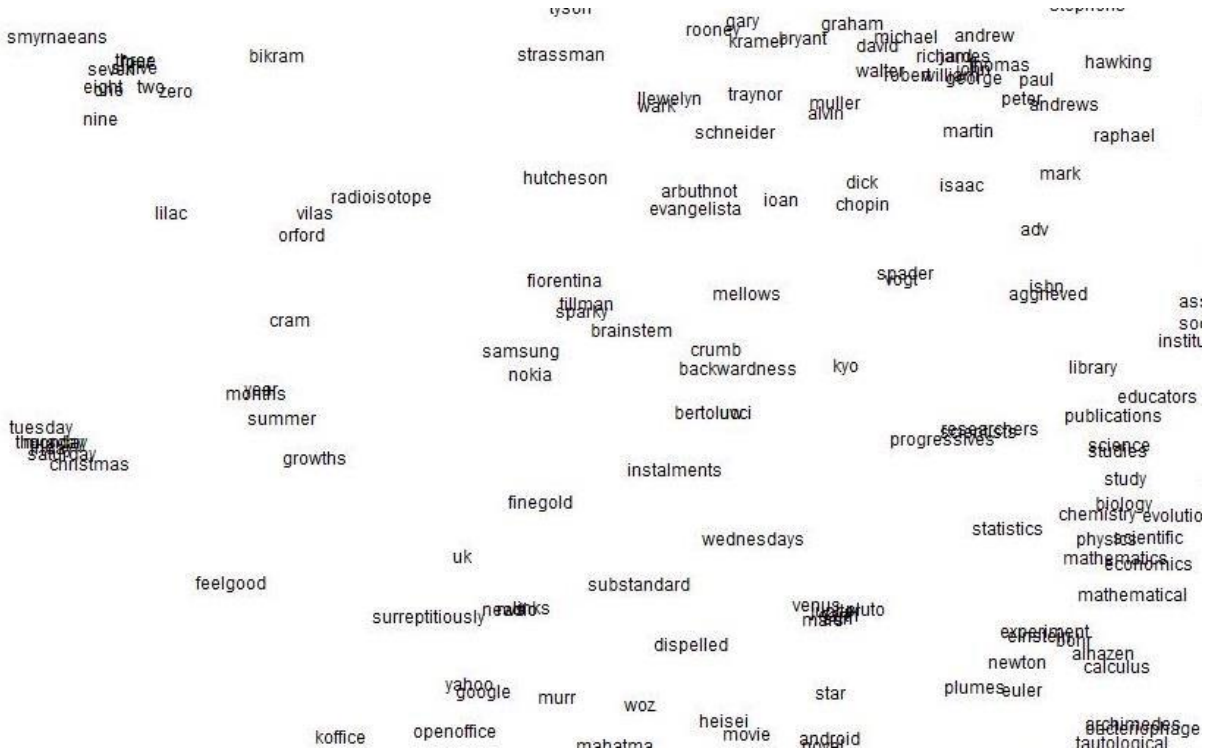


- Objective function

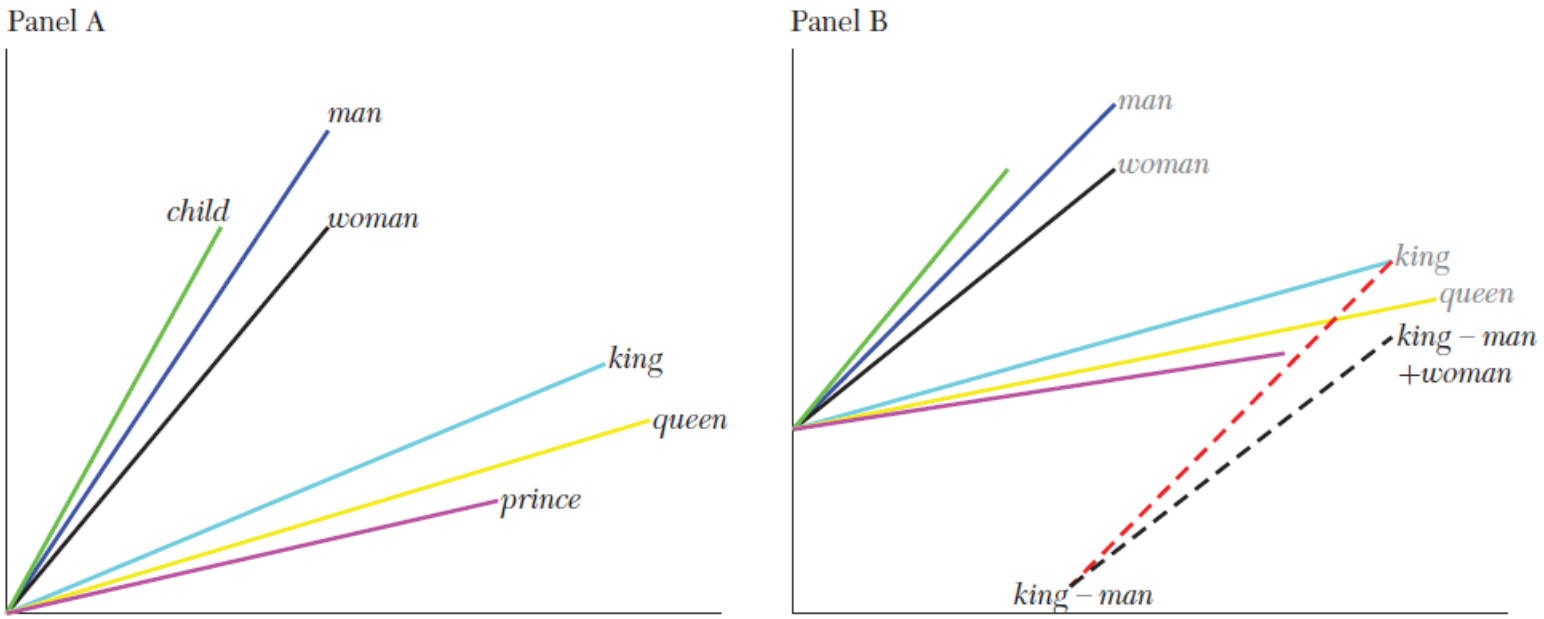
$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\le j\le m\\ j\neq 0}} \log P\left(w_{t+j}\middle|w_t, U, V\right) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\le j\le m\\ j\neq 0}} \log\frac{\exp(u_{t+j}^{T}v_t)}{\sum_{w\in V}\exp(u_w^{T}v_t)}$$

- Solving the optimization problem: Gradient descent and its variants
- **Input embedding (center word embedding, V)**: This is the embedding that represents the word itself
  - The vector that we would use as the final representation of the word after training
- Output Embedding (Context Word Embedding, U): This embedding is used to predict the context words given the center word.
  - It is not typically used as the final representation of the word, but it plays a crucial role during training
- Word2Vec is an unsupervised or self-supervised algorithm

Word2Vec Visualization



Word2Vec Visualization



Word2Vec: Continuous Bag of Words (CBOW)

- CBOW is a variant of skip-gram of Word2Vec. The objective for CBOW is to predict the center words using context words.

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \log P(w_t | w_{t-m}, \dots, w_{t-1}, w_{t+1}, \dots w_{t+m}; U, V)$$

- Estimating the conditional probability $P(w_t | w_{t-m}, \dots, w_{t-1}, w_{t+1}, \dots w_{t+m})$ using softmax

$$P(w_t | w_{t-m}, \dots, w_{t-1}, w_{t+1}, \dots w_{t+m}) = \frac{\exp(u_{w_t}^T \bar{v}_{\text{context}})}{\sum_{w \in V} \exp(u_w^T \bar{v}_{\text{context}})}$$

  where $\bar{v}_{\text{context}}$ is the average of the input embeddings for the context words $w_{t-m}, \dots, w_{t+m}$

- Skip-gram vs. Continuous bag of words
  - **Skip-gram is often preferred** for most NLP tasks because it produces higher-quality embeddings, especially when the dataset is large and rare words are important

Global Vectors for Word Representation (GloVe)

[PDF] **Glove**: Global vectors for word representation

J Pennington, R Socher… - Proceedings of the 2014 …, 2014 - aclanthology.org

… **GloVe**, … **GloVe** model outperforms all other methods on all evaluation metrics, except for the CoNLL test set, on which the HPCA method does slightly better. We conclude that the **GloVe** …

☆ Save Cite Cited by 44618 Related articles All 27 versions »

Global Vectors for Word Representation (GloVe)

- GloVe is an unsupervised or self-supervised learning algorithm for obtaining vector representations (embeddings) for words
- Key ideas behind GloVe: The **ratio of word co-occurrence probabilities** can encode meaningful semantic relationships
  - For example: The ratio of the co-occurrence probabilities of "ice" and "steam" with other words can reveal their relationship
  - "solid" is more related to "ice," while "gas" is more related to "steam"
- How does GloVe work?
  - Constructing the co-occurrence matrix
  - Optimizing the objective function

GloVe: Co-occurrence Matrix

- Denote the co-occurrence matrix as X
  - Each entry X_{ij} represents the number of times word j appears in the context of word i
- Example
  - Consider window length of 1 (more common: 5-10)
  - Corpus: “I like deep learning”, “I like NLP”, “I enjoy flying”
- Compute co-occurrence probabilities

$$P(k|i) = \frac{X_{ik}}{\sum_{j=1}^{V} X_{ij}}$$

	I	like	enjoy	deep	learning	nlp	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
nlp	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

Example Co-occurrence Matrix

GloVe: Objective Function

The objective function is given by:

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

where:

- V: Vocabulary size (number of unique words in the corpus)
- X_{ij}: Co-occurrence count of word j in the context of word i
- w_i: Word vector for word i
- $\tilde{w}_j$: Context vector for word j
- b_i: Bias term for word i
- $\tilde{b}_j$: Bias term for word j
- $f(X_{ij})$: Weighting function to balance the influence of rare and frequent co-occurrences

Decomposing the Objective Function

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

- $w_i^T \tilde{w}_j$: The dot product measures the similarity between the two word vectors
- b_i and $\tilde{b}_j$: Bias terms account for the inherent frequency of words
  - For example, common words like “the” or “and” tend to co-occur with many other words, so their bias terms help adjust for this
  - They ensure that the model can capture the baseline probability of word co-occurrence
- $\log X_{ij}$: Taking the log helps to compress the range of co-occurrence counts, making the optimization process more stable
  - The goal is to make the dot product $w_i^T \tilde{w}_j + b_i + \tilde{b}_j$ approximate $\log X_{ij}$

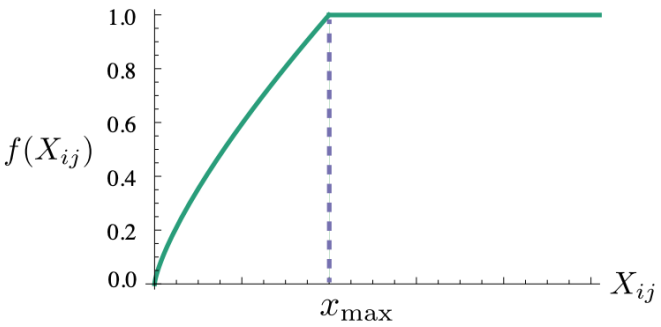
Decomposing the Objective Function (Cont.)

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

- $f(X_{ij})$: A weighting function that controls the influence of each co-occurrence pair (i, j) on the objective function
  - Very frequent co-occurrences (e.g., “the” or “and”) do not dominate the objective.
  - Rare co-occurrences are not ignored.
- A common choice for $f(X_{ij})$ is

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{x_{max}}\right)^{\alpha}, & \text{if } X_{ij} < x_{max} \\ 1, & \text{otherwise} \end{cases}$$

where x_{max} is a pre-specified threshold



Summary

- The GloVe objective function is designed to ensure that the dot product of word vectors and context vectors, adjusted by bias terms, i.e., $w_i^T \tilde{w}_j + b_i + \tilde{b}_j$, approximates the logarithm of their co-occurrence count $\log X_{ij}$
- GloVe captures semantic relationships defined by co-occurrence
  - Words that frequently co-occur (e.g., “ice” and “cold”) will have similar vectors
  - Words that rarely co-occur (e.g., “ice” and “steam”) will have dissimilar vectors
- After training, the sum of the word vectors w_i and context vectors $\tilde{w}_i$ are used as the final word embeddings, i.e., $w_i + \tilde{w}_i$
- Pretrained GloVe embeddings are available for various languages and corpus sizes (e.g., 50, 100, 200, 300 dimensions)
- **Limitation**: The pretrained embeddings are static for each word and cannot handle different contexts
  - Example: **Apple** in “My favorite fruit is an **Apple**” vs. “The star product of **Apple** is iPhone”
  - Solution: Recurrent Neural Networks (RNNs) and Transformer

Application: Measuring Corporate Culture

Measuring Corporate Culture Using NLP

Li, K., Mai, F., Shen, R., & Yan, X. (2021). Measuring corporate culture using machine learning. *Review of Financial Studies*, 34(7), 3265-3315.

- Leverage word embedding models (word2vec) to analyze 209,480 earnings call transcripts, scoring five key cultural values: *innovation*, *integrity*, *quality*, *respect*, and *teamwork*
- For example, the method automatically identifies words, such as *alliance* and *ecosystem*, phrases, such as *win-win*, and even idioms, such as *shoulder to shoulder* and *hand in glove*, as part of the culture dictionary in association with the cultural value of *teamwork*
- Enable us to study how corporate culture correlates with various business outcomes and how it is shaped by major corporate events like mergers and acquisitions (M&As)
  - Corporate culture correlates with business outcomes, including operational efficiency, risk-taking, earnings management, executive compensation design, firm value, and deal making, and that the culture-performance link is more pronounced in bad times
  - Corporate culture is shaped by major corporate events, such as mergers and acquisitions

Data

- Data source: A large corpus of 209,480 earnings call transcripts spanning 2001–2018
  - The analysis focuses on the question-and-answer (QA) sections of these calls because these parts are less scripted and more likely to reflect the managers' true communication about corporate values
- Text preprocessing
  - The transcripts are processed using tools like Stanford CoreNLP for sentence segmentation, lemmatization, and named entity recognition (NER)
  - Phrases (e.g., "shoulder to shoulder") are detected and concatenated (using methods such as the gensim library's phraser module) to ensure that meaningful multiword expressions are treated as single tokens

Word2Vec

- The Word2Vec algorithm is trained based on the financial corpus to learn word representations
- Rationales
  - Traditional bag-of-words representation ignore word order and context
  - Word2vec overcomes this limitation by embedding words into a dense, continuous vector space where semantic similarity is captured by the proximity of vectors
- Outcomes
  - Every word and phrase is converted into a 300-dimensional vector that encapsulates its semantic relationships
  - The cosine similarity between these vectors is used to determine how close a word/phrase is to the target corporate value words

Constructing the Corporate Culture Dictionary

- **Seed Words:** For each of the five cultural values, the study starts with a list of “seed words” derived from earlier literature (notably Guiso, Sapienza, and Zingales, 2015)
  - For a given cultural value (e.g., teamwork), the model computes the average vector from all its seed words
  - This composite vector represents the central concept of that cultural dimension
- **Identifying Related Words:** The cosine similarity is then calculated between this average seed vector and the vector of every other word in the corpus
  - The top 500 words with the highest similarity scores are automatically selected to form an expanded “culture dictionary” for that value
- **Manual Curation:** Finally, the authors manually inspect the generated dictionary to remove words that are too general, off-context, or simply errors from the automated process

Key Contributions of the Methodology

- **Contextual Nuance:** The use of word2vec allows for capturing the subtle, context-dependent meanings of words and phrases, which is particularly important for a complex and multifaceted concept like corporate culture
- **Scalability:** By automating the extraction of relevant vocabulary from a vast dataset, the approach is highly scalable and adaptable to changes in business language over time
- **Semisupervised Learning:** The methodology strikes a balance between unsupervised learning (allowing the data to “speak for itself”) and supervised guidance (through the use of seed words), making it both flexible and reliable for capturing predefined attributes such as cultural values

Acknowledgement

The lecture note has benefited from various resources, including those listed below. Please contact Zonghao Yang (zyang99@stevens.edu) with any questions or concerns about the use of these materials.

- Lecture Notes on Word Vectors from CS224N 2025 Winter at Stanford University

STEVENS
INSTITUTE OF TECHNOLOGY
1870

THANK **YOU**

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030