MODULE 3 • CS24351

Module 3: Vector Semantics, Distributed Representations & Word Embeddings

Topics 25 to 34 • TF-IDF, PPMI, Word2Vec CBOW/SGNS, GloVe, FastText & Evaluation

Module 3: Vector Semantics, Distributed Representations & Word Embeddings

📌 **Syllabus & Pedagogical Blueprint (Topics 25 to 34)**

Localist (One-Hot) vs. Distributed Semantic Representations • Vector Space Models (VSM) • Term-Document & Term-Context Co-occurrence Matrices • Pointwise Mutual Information (PMI) & Positive PMI (PPMI) • TF-IDF Weighting Mechanics • Cosine Semantic Proximity Metric • Word2Vec Architectures: Continuous Bag of Words (CBOW) & Skip-Gram with Negative Sampling (SGNS) • Mathematical Derivation of Negative Sampling Objective • Hierarchical Softmax & Huffman Coding Trees • GloVe (Global Vectors for Word Representation) Log-Bilinear Least Squares Objective • FastText Subword Character N-gram Architecture • Intrinsic Evaluation (WordSim-353, SimLex-999, Word Vector Arithmetic Analogies) vs. Extrinsic Downstream Task Evaluation • 15 Solved University Examination Questions.

Topic 25: The Distributional Hypothesis & Representation Paradigms

The foundational axiom of modern statistical semantics is Firth's (1957) **Distributional Hypothesis**: *"You shall know a word by the company it keeps."* Words occurring in identical linguistic contexts share similar semantic properties.

Representation	Dimensionality & Geometry	Key Computational Limitation
One-Hot Encoding	Sparse $\|V\|$-dimensional binary vector with single 1 at word index (e.g., $[0, 0, 1, 0, \dots]^T$).	Orthogonal vectors: $\mathbf{x}_{\text{cat}}^T \mathbf{x}_{\text{dog}} = 0$. Zero inherent semantic similarity; massive curse of dimensionality.
Count-Based VSM (TF-IDF / PPMI)	Sparse, high-dimensional co-occurrence matrix transformed via statistical association metrics.	Requires SVD dimensionality reduction; computationally expensive $O(\|V\|^2)$ scaling.
Distributed Dense Embeddings (Word2Vec, GloVe)	Dense, low-dimensional real-valued vectors $\mathbb{R}^d$ ($d \in [100, 300]$).	Learned via self-supervised continuous neural optimization; captures geometric analogies ($\mathbf{v}_{\text{King}} - \mathbf{v}_{\text{Man}} + \mathbf{v}_{\text{Woman}} \approx \mathbf{v}_{\text{Queen}}$).

Topic 26: Pointwise Mutual Information (PMI) & TF-IDF

Given word w and context word c, Pointwise Mutual Information measures how much more often they co-occur than expected under random chance:

$$\text{PMI}(\mathbf{w}, \mathbf{c}) = \log_2 \frac{\mathbf{P}(\mathbf{w}, \mathbf{c})}{\mathbf{P}(\mathbf{w})\mathbf{P}(\mathbf{c})} = \log_2 \frac{\mathbf{C}(\mathbf{w}, \mathbf{c}) \cdot \mathbf{N}}{\sum_{\mathbf{c}'} \mathbf{C}(\mathbf{w}, \mathbf{c}') \sum_{\mathbf{w}'} \mathbf{C}(\mathbf{w}', \mathbf{c})}$$

$$\text{Positive PMI (PPMI)}: \text{PPMI}(\mathbf{w}, \mathbf{c}) = \max(\mathbf{0}, \text{PMI}(\mathbf{w}, \mathbf{c}))$$

Topic 27: Word2Vec — Continuous Bag of Words (CBOW) & Skip-Gram

Mikolov et al. (2013) proposed two complementary shallow two-layer neural architectures for learning dense word embeddings from large unannotated corpora:

🧠 **CBOW vs. Skip-Gram Architectures**

- **CBOW (Continuous Bag of Words):** Predicts the target center word w_t given context words within window c: $[w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}]$. Context vectors are averaged: $\mathbf{h} = \frac{1}{2c} \sum \mathbf{v}_{w_{t+j}}$. Fast training; excellent representation for frequent words.
- **Skip-Gram:** Predicts the surrounding context words given the center word w_t. Slower, but superior for small datasets and rare words.

Mathematical Formulation of Skip-Gram with Negative Sampling (SGNS)

Standard softmax requires computing $\sum_{w=1}^{|V|} \exp(\mathbf{v}'_w \cdot \mathbf{v}_c)$ across millions of vocabulary words—a prohibitive computational bottleneck. **Negative Sampling** re-frames multi-class classification as binary logistic regression:

$$\mathcal{L}_{\text{SGNS}}(\theta) = \sum_{\mathbf{t=1}}^{\mathbf{T}} \sum_{-\mathbf{c} \le \mathbf{j} \le \mathbf{c}, \mathbf{j} \neq \mathbf{0}} \left[\log \sigma(\mathbf{v}'_{\mathbf{w_{t+j}}} \cdot \mathbf{v}_{\mathbf{w_t}}) + \sum_{\mathbf{k=1}}^{\mathbf{K}} \mathbb{E}_{\mathbf{w_{n_k}} \sim \mathbf{P_n(w)}} \left[\log \sigma(-\mathbf{v}'_{\mathbf{w_{n_k}}} \cdot \mathbf{v}_{\mathbf{w_t}}) \right] \right]$$

$$\text{where } \sigma(z) = \frac{1}{1+e^{-z}}, \quad \text{Noise Distribution: } P_n(w) = \frac{U(w)^{3/4}}{\sum_{w'} U(w')^{3/4}}$$

The unigram exponent $3/4$ raises the relative sampling probability of rare words relative to ubiquitous stop words.

Topic 28: GloVe — Global Vectors for Word Representation

Pennington et al. (2014) combined the advantages of global matrix factorization and local context window methods. GloVe observes that semantic ratios of co-occurrence probabilities encode linguistic properties:

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

$$\text{Weighting Function: } f(x) = \begin{cases} (x/x_{\max})^\alpha & \text{if } x < x_{\max} \quad (\alpha = 0.75, x_{\max} = 100) \\ 1 & \text{otherwise} \end{cases}$$

Topic 29: FastText — Subword Character N-Grams

Bojanowski et al. (2017) extended Skip-Gram by representing each word w as a set of character n-grams bounded by `<` and `>`:

$$\text{For word } \text{"where"} \text{ (n=3): } \{\text{<wh}, \text{whe}, \text{her}, \text{ere}, \text{re>}\}$$

$$\mathbf{v}_w = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g$$

This enables FastText to compute robust vector representations for **Out-Of-Vocabulary (OOV) words** (e.g., misspelled words or rare biomedical terms) by summing the learned vectors of their constituent subword n-grams!

Topic 30: Master University Examination Question Bank (Module 3)

Q1. Explain Hierarchical Softmax in Word2Vec and how it reduces complexity from $O(|V|)$ to $O(\log_2 |V|)$. (8 Marks)

Hierarchical Softmax (Morin & Bengio 2005) represents the output vocabulary as the leaf nodes of a binary Huffman coding tree where frequent words have shorter branch depths. Instead of evaluating $|V|$ output logits, the model evaluates the probability of taking a left vs. right branch at each internal node n along the root-to-leaf path $p(w) = (n_1, \ldots, n_L)$:

$$P(w \mid w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \text{ch}(n(w,j))]\!] \cdot \theta_j^T \mathbf{v}_{w_I} \right)$$

Because the average depth in a Huffman tree is $O(\log_2 |V|)$, training complexity drops exponentially from $O(10^6)$ to ≈ 20 operations per token!

Q2. Explain Intrinsic vs. Extrinsic Evaluation of Word Embeddings. (8 Marks)

• **Intrinsic Evaluation:** Evaluates word vector quality directly on isolated semantic benchmark tasks (WordSim-353, SimLex-999 correlation with human semantic similarity ratings; syntactic/semantic word analogy accuracy: $a : b :: c : d \implies \arg\max_x \cos(\mathbf{v}_x, \mathbf{v}_b - \mathbf{v}_a + \mathbf{v}_c)$). Fast and model-agnostic, but high intrinsic scores do not guarantee downstream task gains.
• **Extrinsic Evaluation:** Plugs word embeddings as frozen/fine-tuned feature extractors into real-world downstream NLP pipelines (Named Entity Recognition, Sentiment Classification, Machine Translation) and measures end-task metrics (F1-score, BLEU, Accuracy). Computationally expensive and task-specific.

Q3. Detail the Cosine Similarity Metric and why Euclidean Distance is inappropriate for raw text term vectors. (6 Marks)

$$\text{Cosine Similarity: } \cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$$

Euclidean distance $\|\mathbf{u} - \mathbf{v}\|_2$ is severely confounded by document length—a 10-page document discussing Biology and a 1-paragraph summary will have massive Euclidean distance purely due to scalar term frequency magnitudes. Cosine similarity normalizes vector lengths to unit norm (L_2), measuring strictly the angle between semantic directions independent of document length!

Topic 31: Mathematical Proof & Gradient Derivation of Word2Vec SGNS

For a single center word c and true context word o with K negative samples $\{n_1, \ldots, n_K\}$ drawn from $P_n(w)$:

$$\mathcal{L} = \log \sigma(\mathbf{u_o^T v_c}) + \sum_{\mathbf{k=1}}^{\mathbf{K}} \log \sigma(-\mathbf{u_{n_k}^T v_c})$$

Derivation of Parameter Gradients with Respect to Word Vectors

Recall that $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ and $\frac{d}{dz} \log \sigma(z) = 1 - \sigma(z)$, while $\frac{d}{dz} \log \sigma(-z) = -\sigma(z)$.

1. Gradient with respect to true context vector $\mathbf{u}_o$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u_o}} = (\mathbf{1} - \sigma(\mathbf{u_o^T v_c}))\mathbf{v_c}$$

2. Gradient with respect to negative context vector $\mathbf{u}_{n_k}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u_{n_k}}} = -\sigma(\mathbf{u_{n_k}^T v_c})\mathbf{v_c}$$

3. Gradient with respect to center word vector $\mathbf{v}_c$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v_c}} = (\mathbf{1} - \sigma(\mathbf{u_o^T v_c}))\mathbf{u_o} - \sum_{\mathbf{k=1}}^{\mathbf{K}} \sigma(\mathbf{u_{n_k}^T v_c})\mathbf{u_{n_k}}$$

Interpretation: Gradient descent pushes the center vector $\mathbf{v}_c$ *towards* the true context vector $\mathbf{u}_o$ while repelling it *away* from all K negative samples simultaneously!

Topic 32: Latent Semantic Analysis (LSA) & SVD Dimensionality Reduction

LSA (Deerwester et al. 1990) factorizes a Term-Document co-occurrence matrix $\mathbf{X} \in \mathbb{R}^{|V| \times |D|}$ using Singular Value Decomposition:

$$\mathbf{X} \approx \mathbf{U_k \Sigma_k V_k^T}$$

- $\mathbf{U}_k \in \mathbb{R}^{|V| \times k}$: Dense k-dimensional word representation vectors.
- $\mathbf{\Sigma}_k \in \mathbb{R}^{k \times k}$: Diagonal matrix of top k singular values (strengths of latent topics).
- $\mathbf{V}_k^T \in \mathbb{R}^{k \times |D|}$: Dense k-dimensional document representation vectors.

- *Limitation:* Computationally cubic $O(|V| \cdot |D|^2)$ time, and fails to capture non-linear contextual semantics.

Topic 34: Advanced Semantic Embeddings & University Exam Problem Bank (Part II)

🏛️ Solved Numerical 11: Complete Pointwise Mutual Information (PMI) Matrix Computation

Consider a 3×3 word-context co-occurrence matrix from a mini-corpus ($N = 100$ total co-occurrences):

Target \ Context	`computer`	`data`	`pie`	Row Sum $C(w)$
`digital`	20	15	0	35
`cherry`	0	0	25	25
`information`	10	30	0	40
Col Sum $C(c)$	30	45	25	$\mathbf{N = 100}$

1. Compute PMI(`digital`, `computer`):

$$P(\text{digital}, \text{computer}) = \frac{20}{100} = 0.20 \qquad P(\text{digital}) = \frac{35}{100} = 0.35 \qquad P(\text{computer}) = \frac{30}{100} = 0.30$$

$$\text{PMI} = \log_2\left(\frac{\mathbf{0.20}}{\mathbf{0.35 \times 0.30}}\right) = \log_2\left(\frac{\mathbf{0.20}}{\mathbf{0.105}}\right) = \log_2(\mathbf{1.9048}) = +\mathbf{0.9296}$$

2. Compute PMI(`digital`, `pie`) with Zero Count:

$$\text{PMI} = \log_2(0) = -\infty \implies \text{PPMI}(\text{digital}, \text{pie}) = \max(\mathbf{0}, -\boldsymbol{\infty}) = \mathbf{0.0000}$$

Q4. Explain Why Word2Vec Uses the Subsampling of Frequent Words and How the Probability Threshold is Formulated. (8 Marks)

In raw natural text, ubiquitous words like `the`, `of`, `and` occur millions of times, providing near-zero semantic discrimination while consuming massive training compute. Mikolov et al. subsample words during training, discarding word w_i with probability:

$$\mathbf{P}(\text{discard } \mathbf{w_i}) = \mathbf{1} - \sqrt{\frac{\mathbf{t}}{\mathbf{f(w_i)}}}$$

Where $f(w_i)$ is the corpus frequency fraction of word w_i and $t \approx 10^{-4}$ is a threshold parameter. Words with frequency $f(w) > t$ are discarded with high probability, accelerating training speed by $2\times$ to $10\times$ while drastically improving vector representations of rare words!

Topic 34.3: Master University Exam Problem Bank (Part III)

🏛️ Solved Problem 12: Continuous Bag of Words (CBOW) Hidden-to-Output Weight Update

In CBOW, let context words be $\{w_1, \dots, w_{2C}\}$ and true target center word be w_t (j^*).

- Hidden representation: $\mathbf{h} = \frac{1}{2C} \sum_{i=1}^{2C} \mathbf{v}_{w_i} \in \mathbb{R}^d$.
- Output logits: $z_j = \mathbf{u}_j^T \mathbf{h}$ for all vocabulary words $j \in \{1, \dots, |V|\}$.
- Predicted probability via softmax: $\hat{y}_j = \frac{\exp(\mathbf{u}_j^T \mathbf{h})}{\sum_k \exp(\mathbf{u}_k^T \mathbf{h})}$.
- Cross-entropy loss: $\mathcal{L} = -\log \hat{y}_{j^*}$.
- Output weight gradient: $\frac{\partial \mathcal{L}}{\partial \mathbf{u}_j} = (\hat{y}_j - y_j)\mathbf{h}$ where $y_j = 1$ if $j = j^*$ else 0.
- Input weight gradient for each context word w_i: $\frac{\partial \mathcal{L}}{\partial \mathbf{v}_{\mathbf{w_i}}} = \frac{1}{2C} \sum_{j=1}^{|V|} (\hat{\mathbf{y}}_\mathbf{j} - \mathbf{y}_\mathbf{j}) \mathbf{u}_\mathbf{j}$.

Q5. Explain the Geometric Intuition of Word Vector Analogies and the Cosine Offset Method. (8 Marks)

Word vectors trained on large corpora organize semantic relationships as linear directional vector offsets in $\mathbb{R}^d$:

$$\mathbf{v}_{\text{King}} - \mathbf{v}_{\text{Man}} \approx \mathbf{v}_{\text{Queen}} - \mathbf{v}_{\text{Woman}} \implies \mathbf{v}_{\text{King}} - \mathbf{v}_{\text{Man}} + \mathbf{v}_{\text{Woman}} \approx \mathbf{v}_{\text{Queen}}$$

To solve an analogy $a : b :: c : ?$, 3CosAdd finds word d^* maximizing:

$$\mathbf{d}^* = \arg \max_{\mathbf{x} \notin \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}} \cos(\mathbf{v_x}, \mathbf{v_b} - \mathbf{v_a} + \mathbf{v_c}) = \arg \max_{\mathbf{x}} \left[\cos(\mathbf{v_x}, \mathbf{v_b}) - \cos(\mathbf{v_x}, \mathbf{v_a}) + \cos(\mathbf{v_x}, \mathbf{v_c})\right]$$

Levy & Goldberg (2014) introduced **3CosMul** to prevent single large similarity terms from dominating:

$$\mathbf{d}^* = \arg \max_{\mathbf{x}} \frac{\cos(\mathbf{v_x}, \mathbf{v_b}) \cdot \cos(\mathbf{v_x}, \mathbf{v_c})}{\cos(\mathbf{v_x}, \mathbf{v_a}) + \epsilon}$$

Q6. Compare Contextualized Word Representations (BERT, ELMo) vs. Static Word Embeddings (Word2Vec, GloVe). (8 Marks)

Dimension	Static Embeddings (Word2Vec / GloVe)	Contextualized Embeddings (ELMo / BERT)
Polysemy	Single fixed vector per word; `bank` in "river bank" and "financial bank" have identical representations.	Dynamic vector computed as a function of the entire sentence; `bank` has distinct embeddings in different contexts.
Lookup	Fast static matrix row lookup $O(1)$.	Requires deep multi-layer transformer forward pass.
Syntax Integration	Captures only local bag-of-words co-occurrence.	Encodes deep syntactic hierarchical dependency structures and semantic roles.

🏛️ Solved Problem 17: GloVe Log-Bilinear Model Derivation from Co-occurrence Probability Ratios

Let $P_{ik} = P(k \mid i) = \frac{X_{ik}}{X_i}$ be the conditional probability of context word k given word i. Consider probe word $k = \text{"solid"}$ with $i = \text{"ice"}$ and $j = \text{"steam"}$:

- $P(\text{solid} \mid \text{ice}) \approx \text{Large}$, while $P(\text{solid} \mid \text{steam}) \approx \text{Small} \implies \frac{P_{ik}}{P_{jk}} \gg 1.0$.
- For probe word $k = \text{"gas"}$: $\frac{P(\text{gas}|\text{ice})}{P(\text{gas}|\text{steam})} \ll 1.0$.
- For noise word $k = \text{"water"}$: $\frac{P(\text{water}|\text{ice})}{P(\text{water}|\text{steam})} \approx 1.0$.
- To encode this ratio linearly in vector space, the relationship must satisfy homomorphism $F(\mathbf{w}_i - \mathbf{w}_j, \tilde{\mathbf{w}}_k) = \frac{P_{ik}}{P_{jk}}$. Setting $F = \exp$:

$$\mathbf{w_i^T \tilde{w}_k} - \mathbf{w_j^T \tilde{w}_k} = \ln \mathbf{P_{ik}} - \ln \mathbf{P_{jk}} = \ln \mathbf{X_{ik}} - \ln \mathbf{X_i} - (\ln \mathbf{X_{jk}} - \ln \mathbf{X_j})$$

Absorbing $\ln X_i$ into scalar bias terms $b_i, \tilde{b}_k$ yields the canonical GloVe objective:

$$\mathbf{w_i^T \tilde{w}_k} + \mathbf{b_i} + \mathbf{\tilde{b}_k} = \ln \mathbf{X_{ik}}$$

Q7. Detail the FastText Hashing Trick for Storing Millions of Subword N-Grams in Memory. (8 Marks)

In morphologically rich languages, the number of unique character n-grams (sizes 3 to 6) easily exceeds 10 million distinct strings, requiring gigabytes of RAM for embedding lookup tables.

• **The Hashing Trick (Fowler-Noll-Vo / MurmurHash):** Instead of storing a dictionary mapping strings to integers, FastText hashes each character n-gram string directly into a fixed-size integer bucket array of size $K = 2 \times 10^6$ using hash function $h(g) = \text{hash}(g) \pmod K$. Multiple rare subwords map to shared embedding buckets, providing massive parameter compression with negligible accuracy degradation!

🏛️ Solved Problem 19: Word2Vec Negative Sampling Noise Distribution Mathematical Proof

Prove why the unigram distribution raised to the $3/4$ power ($P_n(w) \propto U(w)^{0.75}$) balances frequent and rare negative word sampling:

- Consider rare word A ($U(A) = 0.0001 = 10^{-4}$) and frequent word B ($U(B) = 0.01 = 10^{-2}$).
- Raw ratio: $\frac{U(B)}{U(A)} = \frac{0.01}{0.0001} = \mathbf{100.0}$ (B is sampled $100\times$ more often than A).
- Adjusted counts: $U(A)^{0.75} = (10^{-4})^{0.75} = 10^{-3} = 0.001; U(B)^{0.75} = (10^{-2})^{0.75} = 10^{-1.5} \approx 0.0316$.
- Adjusted ratio: $\frac{U(B)^{0.75}}{U(A)^{0.75}} = \frac{0.0316}{0.001} \approx \mathbf{31.6}$!
- *Result:* Rare word A receives a $3.16\times$ relative boost in negative sampling representation, preventing under-training!

🏛️ Solved Problem 20: Word Embedding Evaluation via Spearman Rank Correlation on WordSim-353

Given 5 word pairs with human similarity rankings $H = [1, 2, 3, 4, 5]$ and model cosine similarity rankings $M = [1, 3, 2, 5, 4]$:

- Rank differences $d_i = H_i - M_i$: $[0, -1, +1, -1, +1] \implies \sum d_i^2 = 0 + 1 + 1 + 1 + 1 = \mathbf{4.0}$.
- $$\rho = \mathbf{1} - \frac{\mathbf{6 \sum d_i^2}}{\mathbf{n(n^2 - 1)}} = \mathbf{1} - \frac{\mathbf{6(4)}}{\mathbf{5(25 - 1)}} = \mathbf{1} - \frac{\mathbf{24}}{\mathbf{120}} = \mathbf{1 - 0.20} = \mathbf{+0.8000}$$
- $$\mathbf{\text{Conclusion: Strong positive correlation (+0.80) with human cognitive semantic judgment!}}$$

🏛️ Solved Problem 28: Word2Vec Continuous Bag of Words (CBOW) Computational Complexity

For training corpus of T words, vocabulary size $|V|$, embedding dimension d, and context window size C:

- Input-to-Hidden projection: $2C \times d$ lookups $\implies O(C \cdot d)$.
- Hidden-to-Output with Standard Softmax: $|V| \times d$ operations $\implies \mathbf{O(T \cdot (C \cdot d + |V| \cdot d))}$.
- Hidden-to-Output with Negative Sampling (K samples): $(K+1) \times d$ operations $\implies \mathbf{O(T \cdot (C \cdot d + K \cdot d))}$.
- $$\text{Speedup with Negative Sampling: } \frac{\mathbf{|V|}}{\mathbf{K}} \approx \frac{\mathbf{100,000}}{\mathbf{5}} = \mathbf{20,000} \times \text{ Faster!}$$

🏛️ Solved Problem 29: Pointwise Mutual Information Matrix Factorization Equivalence

Levy & Goldberg (2014) proved that Word2Vec Skip-Gram with Negative Sampling (SGNS) is implicitly factorizing a shifted Pointwise Mutual Information matrix:

$$\mathbf{W C^T} = \mathbf{M^{PMI}} - \log(\mathbf{K}) \quad \text{where } \mathbf{M_{ij}^{PMI}} = \text{PMI}(\mathbf{w_i}, \mathbf{c_j}) = \log \frac{\mathbf{P(w_i, c_j)}}{\mathbf{P(w_i)P(c_j)}}$$

Where K is the number of negative samples! Bridges neural embeddings directly with classical distributional co-occurrence matrices!

🏛️ Solved Problem 30: GloVe Weighting Exponent $\alpha = 0.75$ Justification

The weighting function $f(X_{ij}) = (X_{ij}/100)^{0.75}$ ensures that extremely frequent stopwords do not dominate the weighted least squares objective while rare co-occurrences are not completely neglected.

🏛️ Solved Problem 39: FastText Subword Character N-Gram Vector Aggregation Trace

Given word $w = \text{"cat"}$ ($n = 3$): Subwords = $\{\text{}, \text{}\}$. Vectors in 2D space:

- $\mathbf{z}_{\text{}} = (0.1, 0.3)^T, \ \mathbf{z}_{\text{}} = (0.6, 0.9)^T$.
- $$\mathbf{v}_{\text{cat}} = \sum_{\mathbf{g}} \mathbf{z_g} = (\mathbf{0.2 + 0.5 + 0.1 + 0.6},\ \mathbf{0.4 + 0.8 + 0.3 + 0.9})^{\mathbf{T}} = (\mathbf{1.4, 2.4})^{\mathbf{T}}$$
- For unseen misspelled word $w' = \text{"cats"}$ with subwords $\{\text{}, \text{}\}$: $Even if `cats` was never in the training dictionary, the model computes a rich vector using shared subwords$\text{

🏛️ Solved Problem 40: Term Frequency–Inverse Document Frequency (TF–IDF) Weighting Trace

In a collection of $N = 10,000$ scientific documents, term $t = \text{"transformer"}$ appears 15 times in a 300-word document D_1, and occurs in 100 total documents across the corpus:

$$\text{TF}(t, D_1) = \frac{15}{300} = \mathbf{0.05} \qquad \text{IDF}(t) = \log_{10}\left(\frac{10,000}{100}\right) = \log_{10}(100) = \mathbf{2.0}$$

$$\text{TF-IDF}(\mathbf{t}, \mathbf{D_1}) = \mathbf{0.05} \times \mathbf{2.0} = \mathbf{0.1000}$$

🏛️ Solved Problem 48: Skip-Gram with Hierarchical Softmax Binary Tree Encoding Trace

In a 4-word vocabulary $V = \{\text{the}, \text{cat}, \text{sat}, \text{mat}\}$ structured as a balanced binary tree:

- Target word `mat` path: Root $\xrightarrow{n_1(\text{Right: } 1)} n_2 \xrightarrow{n_2(\text{Left: } 0)} \text{Leaf}(\text{mat})$.
- $P(\text{mat} \mid \text{cat}) = \sigma(\theta_{n_1}^T \mathbf{v}_{\text{cat}}) \times (1 - \sigma(\theta_{n_2}^T \mathbf{v}_{\text{cat}}))$.
- Zero negative sampling required; evaluated in exactly 2 dot products ($O(\log_2 4) = 2$)!

🏛️ Solved Problem 49: Contextual Similarity via Word-to-Sentence Earth Mover's Distance (WMD)

Word Mover's Distance (Kusner et al. 2015) measures the minimum cumulative semantic distance required to transport the embedded word vectors of sentence S_1 to sentence S_2 via optimal transport linear programming!

🏛️ Solved Problem 56: Skip-Gram with Negative Sampling (SGNS) Computational Speedup Proof

Prove that computing binary logistic loss on $K = 5$ negative samples reduces GPU gradient backpropagation time by $\mathbf{99.995\%}$ compared to full vocabulary softmax ($|V| = 100,000$)!

🏛️ Solved Problem 57: Contextual Polysemy Disambiguation in Transformer Self-Attention

In sentence $S_1 = \text{"The bank riverbed was eroded"}$ vs $S_2 = \text{"The bank approved the loan"}$, trace how self-attention dynamically projects the token `bank` into divergent subspace coordinates!

🏛️ Solved Problem 64: Continuous Bag of Words (CBOW) vs Skip-Gram Negative Sampling Convergence

CBOW averages context embeddings, smoothing out stochastic noise and converging in $3\times$ fewer epochs on large dense corpora. Skip-Gram treats each (target, context) pair independently, giving rare words repeated training updates!

🏛️ Solved Problem 65: GloVe Weighting Function Optimization Properties

Show that when $X_{ij} \geq x_{\max} = 100$, $f(X_{ij}) = 1.0$ caps the loss contribution, preventing extremely frequent pairs like `(of, the)` from warping semantic distances!

🏛️ Solved Problem 71: Continuous Bag of Words (CBOW) Projection Layer Gradient Proof

Show that averaging context words $\mathbf{h} = \frac{1}{2C} \sum \mathbf{v}_{w_{t+j}}$ distributes the output error gradient equally across all active context window representations $\frac{\partial \mathcal{L}}{\partial \mathbf{v}_c} = \frac{1}{2C} \mathbf{e}_{\text{output}}$!

🏛️ Solved Problem 75: Word Vector Semantic Linear Subspace Orthogonality Proof

Prove that in a well-trained 300D GloVe embedding space, the gender direction $\mathbf{g} = \mathbf{v}_{\text{woman}} - \mathbf{v}_{\text{man}}$ is approximately orthogonal to the grammatical tense direction $\mathbf{t} = \mathbf{v}_{\text{walked}} - \mathbf{v}_{\text{walk}}$: $\mathbf{g}^T \mathbf{t} \approx 0.012$, proving modular disentanglement of semantic and syntactic features!

🏛️ Solved Problem 79: Cosine Distance vs Angular Distance in Semantic Geometry

Angular distance $d_\theta(\mathbf{u}, \mathbf{v}) = \frac{\arccos(\cos(\mathbf{u}, \mathbf{v}))}{\pi}$ satisfies all metric space axioms (including the Triangle Inequality), unlike raw cosine distance $(1 - \cos(\mathbf{u}, \mathbf{v}))$ which is a semi-metric!

🏛️ Solved Problem 80: Word2Vec Vector Arithmetic Stability Analysis

Show that subtracting $\mathbf{v}_{\text{man}}$ removes gender attributes from the masculine subspace while retaining royalty features: $\mathbf{v}_{\text{king}} - \mathbf{v}_{\text{man}} \approx \mathbf{v}_{\text{royalty}}$, which upon adding $\mathbf{v}_{\text{woman}}$ lands precisely in the feminine royal semantic neighborhood of $\mathbf{v}_{\text{queen}}$!

🏛️ Solved Problem 86: FastText vs Word2Vec Out-Of-Vocabulary Robustness Proof

For unseen neologism *"biocomputational"*: Word2Vec assigns random `` vector. FastText decomposes into subwords $\{\text{}\}$, summing subword vectors to land precisely at the geometric intersection of biology and computer science!

🏛️ Solved Problem 92: Word Embedding Intrinsic Analogies Benchmark Evaluation

On the Google Analogy Test Set (19,544 questions across syntactic and semantic categories), Skip-Gram with Negative Sampling ($d = 300$) achieves $\mathbf{74.2\%}$ accuracy, outperforming classical LSA by over 45%!

🏛️ Solved Problem 98: GloVe Least-Squares Global Gradient Derivation

The gradient of GloVe loss J with respect to target word vector $\mathbf{w}_i$ is: $\frac{\partial J}{\partial \mathbf{w}_i} = 2 \sum_{j=1}^{|V|} f(X_{ij})(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \ln X_{ij}) \tilde{\mathbf{w}}_j$.

🏛️ Solved Problem 102: FastText Subword Matrix Hashing Collisions

Show that with bucket size $B = 2 \times 10^6$ and $N = 3 \times 10^5$ distinct n-grams, the expected collision rate $\frac{N(N-1)}{2B} \approx 0.0225$ has near-zero impact on semantic retrieval accuracy!

🏛️ Solved Problem 105: Pointwise Mutual Information Bias Correction

Standard PMI is heavily biased towards low-frequency words. Positive PMI with frequency exponent $\alpha = 0.75$ ($\text{PPMI}_\alpha(w, c) = \max(0, \log_2 \frac{P(w,c)}{P(w)P_\alpha(c)})$) raises context distribution to 0.75, mitigating rare word inflation!

🏛️ Solved Problem 108: SVD Singular Value Power Law Decay Trace

In LSA, singular values $\sigma_1, \ldots, \sigma_k$ exhibit exponential power-law decay ($\sigma_i \propto i^{-\alpha}$), proving that retaining top $k = 300$ singular vectors captures $> 90\%$ of semantic information variance!

🏛️ Solved Problem 109: Word2Vec Continuous Skip-Gram Softmax Normalization Bottleneck

Calculating full softmax denominator $\sum_{w=1}^{|V|} \exp(\mathbf{u}_w^T \mathbf{v}_c)$ requires iterating over all 100,000 words per training token, demanding billions of GPU floating point operations per sentence!

🏛️ Solved Problem 114: Word2Vec Continuous Skip-Gram Vector Norm Invariance

Show that cosine similarity between two unit-normalized word vectors equals their inner product $\cos(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}, \mathbf{v} \rangle$, converting vector search into blazing-fast matrix-vector multiplication!

🏛️ Solved Problem 115: FastText Out-Of-Vocabulary Subword Representation Synthesis

For unseen medical neologism *"neurocomputational"*: FastText decomposes into subwords $\{\text{}\}$, summing subword vectors to land precisely at the intersection of neuroscience and computing!