Skip-Gram Variant Evaluation

(Efficient Estimation of Word Representations in Vector Space
Word2Vec – Mikolov et al., 2013)

Abdul Khayyum Farooqui

***Abstract*—This report compares distributional word representations learned by Skip-Gram models with Hierarchical Softmax and Negative Sampling against a TF-IDF baseline under a shared vocabulary and consistent preprocessing. Models are evaluated on semantic similarity, analogy reasoning, clustering quality, and computational efficiency, with additional analysis of frequency effects on the learned representations. Skip-Gram embeddings consistently outperform TF-IDF across all semantic evaluations. Subsampling of frequent words improves both efficiency and embedding quality, with the strongest gains observed for Negative Sampling. Geometric analyses show that Skip-Gram embeddings are less dominated by word frequency and exhibit more structured representation spaces, helping explain their superior semantic performance.**

I. Introduction

Learning meaningful word representations from raw text is a foundational problem in natural language processing. Many downstream tasks such as semantic similarity, clustering, and analogy reasoning depend on representations that capture semantic relationships rather than surface-level co-occurrence. Traditional count-based methods such as TF-IDF are effective for document retrieval but are limited in their ability to model word meaning, as they primarily encode frequency and lexical overlap.

Neural word embedding models address this limitation by learning continuous vector representations in which semantic relationships emerge from geometric structure. The Skip-Gram model introduced by Mikolov et al. learns embeddings by predicting surrounding context words, enabling the capture of syntactic and semantic regularities directly from unlabeled text. Subsequent extensions, including Hierarchical Softmax, Negative Sampling, and subsampling of frequent words, were proposed to improve both training efficiency and representation quality.

This project re-implements the Skip-Gram architecture introduced by Mikolov et al. (2013a) and extends it with Hierarchical Softmax, Negative Sampling, and subsampling of frequent words as proposed in Mikolov et al. (2013b). The Skip-Gram model is trained using both Hierarchical Softmax and Negative Sampling objectives, each evaluated with and without subsampling. A TF-IDF model is included as a baseline to contrast predictive embeddings with traditional frequency-based representations. All models are trained using a shared vocabulary and consistent preprocessing and are evaluated across semantic quality, geometric structure, and computational efficiency. The objective is to understand how training objectives and frequency-based regularization jointly influence the quality and practicality of learned word representations.

II. Problem Formulation

Let the corpus consist of tokenized sentences, where each sentence is a sequence of words drawn from a fixed vocabulary V of size $|V|$. For a sentence $(w_1, w_2, \ldots, w_T)$, the Skip-Gram model constructs training data by pairing each center word w_t with surrounding context words within a symmetric window. A dynamic window is used: for each position t, a radius R is sampled uniformly from $\{1, \ldots, c\}$, and all words w_{t+j} with $j \in [-R, R] \setminus \{0\}$ are treated as context words.

Each word $w \in V$ is associated with two vector representations: an input embedding $v_w \in \mathbb{R}^d$ and an output embedding $u_w \in \mathbb{R}^d$, where $d = 200$ is the embedding dimension. Given a set D of observed center–context pairs, the Skip-Gram objective is to learn embeddings that maximize the conditional likelihood of observing context words given center words. The training objective is

$$\max \sum_{(w,c)\in D} \log p(c \mid w).$$

The conditional probability $p(c \mid w)$ is defined using a softmax over the vocabulary, which is approximated using two alternative objectives. Hierarchical Softmax replaces the full softmax with a binary tree over the vocabulary and models $p(c \mid w)$ as the product of logistic probabilities along the path from the root to the target word. Negative Sampling instead reformulates training as a binary classification problem, where observed center–context pairs are treated as positive examples and contrasted against K negative samples drawn from a noise distribution proportional to the unigram frequency raised to the power 0.75. The model then optimizes a logistic loss over positive and negative pairs.

Subsampling of frequent words is optionally applied during pair construction. Each word is retained with a probability that decreases with its corpus frequency, reducing the influence of very common tokens and modifying the effective training distribution. When subsampling is disabled, all tokens are retained.

Overall, the Skip-Gram model learns word embeddings by optimizing either the Hierarchical Softmax or Negative Sampling objective over observed context relationships. The resulting embeddings are evaluated by their ability to capture semantic similarity, relational structure, and global organization in the learned representation space.

III. Models and Methods

This project implements the Skip-Gram architecture introduced by Mikolov et al. (2013a) and evaluates two training objectives for approximating the full softmax: Hierarchical Softmax and Negative Sampling. Each objective is evaluated both with and without subsampling of frequent words. A TF-IDF model is included as a non-neural baseline. All models are trained on the same tokenized corpus using a shared vocabulary and consistent preprocessing to ensure controlled comparisons.

A. Data Processing and Vocabulary

Text is normalized using lowercasing, removal of URLs and extraneous punctuation, and rule-based sentence segmentation. Tokens are extracted with a deterministic tokenizer that preserves internal apostrophes and hyphens. A minimum frequency threshold of 10 is applied, and the vocabulary is capped at 100,000 types. For neural models, center–context pairs are generated using a symmetric dynamic window with maximum radius $c = 5$.

B. Skip-Gram Architecture

In Skip-Gram, each word $w \in V$ is represented by an input embedding $v_w \in \mathbb{R}^d$, and each context word c is associated with an output embedding $u_c \in \mathbb{R}^d$, where the embedding dimension is $d = 200$. Training operates on explicit center–context pairs (w, c) extracted from the corpus. The learned input embeddings are used for all downstream evaluations.

C. Hierarchical Softmax

Hierarchical Softmax replaces the full softmax over the vocabulary with a binary tree constructed using Huffman coding, so that frequent words have shorter paths. The probability $p(c \mid w)$ is modeled as the product of logistic probabilities along the root-to-leaf path corresponding to the context word. In the implementation, each internal tree node has a learned vector, and the loss is computed as a binary cross-entropy over the sequence of path decisions, vectorized across batches for efficiency.

D. Negative Sampling

Negative Sampling reframes training as a binary classification problem. For each observed center–context pair, the model samples $K = 10$negative context words and learns to assign higher scores to the positive pair than to the negatives. Negative samples are drawn from a noise distribution proportional to the unigram frequency raised to the power 0.75, following Mikolov et al. (2013b). Training optimizes a binary cross-entropy loss over one positive and multiple negative examples using a separate output embedding matrix.

E. Subsampling of Frequent Words

Subsampling is optionally applied during corpus processing to reduce the dominance of extremely frequent tokens. Each word is retained with a probability that decreases with its empirical frequency, using the standard Mikolov-style keep probability with threshold $t = 10^{-4}$. When subsampling is disabled, all tokens are retained. Subsampling is applied before generating training pairs, directly affecting both the effective training distribution and the number of updates.

F. TF-IDF Baseline and Training Details

The TF-IDF baseline uses the same shared vocabulary and produces L2-normalized sparse vectors. All neural models use a 95/5 train–validation split, batch size 256, learning rate 5×10^{-3}, learning-rate scheduling, early stopping, and a fixed random seed. Models are implemented in PyTorch with sparse updates and include detailed system-level monitoring of CPU usage, memory consumption, and runtime to support efficiency comparisons.

IV. Experimental Setup

A. Dataset

All models are trained on a single merged corpus built from two public datasets: Plain Text Wikipedia (Simple English) and A Million News Headlines. The Wikipedia portion contributes simplified, longer-form explanatory text, while the headlines contribute short, information-dense sentences. Using the same merged corpus for every run ensures that differences in results come from the training objective and subsampling, not from data exposure.

B. Preprocessing and tokenization

The two sources are concatenated into one text file with one document per line. Text is Unicode-normalized and lowercased, with URLs, emails, and bracketed content removed. Punctuation is restricted so that sentence boundaries remain identifiable (., ?, !), while tokenization preserves internal apostrophes and hyphens (for example, she's, covid-19). Sentences are segmented by splitting on sentence-ending punctuation and line breaks. If sentence boundaries are unusually sparse, the pipeline falls back to fixed 100-word chunks to avoid unrealistically large context windows. The final output is a tokenized corpus (list of token sequences) shared by all models.

C. Model configurations and hyperparameters

All Skip-Gram variants use embedding dimension $d = 200$ and a dynamic window with maximum radius $c = 5$. A shared vocabulary is constructed once from the tokenized corpus by applying `min_count = 10` and `max_vocab = 100,000`, and that same vocabulary is reused across all Skip-Gram runs and the TF-IDF baseline for comparability. Each Skip-Gram model is trained for 10 epochs. For Negative Sampling, we use $K = 10$ negatives per positive pair sampled from a unigram distribution proportional to $\text{count}(w)^{0.75}$. Subsampling is evaluated as an explicit experimental factor: when enabled, it uses threshold $t = 1e{-}4$, and when disabled it is turned off by setting the threshold to `None`. Aside from the objective (HS vs NS) and subsampling (on vs off), settings are held fixed to support fair comparisons.

D. Baseline

A TF-IDF baseline is trained on the same tokenized corpus using the same fixed vocabulary. The representation uses TF-IDF weighting with IDF smoothing and L2 normalization. This baseline is included to contrast predictive embedding learning against a sparse, frequency-based representation under matched vocabulary coverage.

E. Evaluation protocol and metrics

Models are evaluated using complementary measures of semantic quality, structure, and efficiency. Semantic similarity is measured on WordSim-353 using cosine similarity and Spearman rank correlation against human judgments. Relational structure is evaluated on the Google Word Analogy dataset using standard vector arithmetic with top-1 accuracy, skipping out-of-vocabulary questions. Global structure is evaluated via clustering into five semantic categories (animals, cities, countries, professions, sports) and reporting NMI and ARI against ground-truth labels. Embedding geometry is inspected using 2D PCA and t-SNE visualizations, and further quantified using norm–frequency regression and hubness statistics. Efficiency is measured using logged runtime and throughput, along with recorded CPU and memory utilization during training.

V. Empirical Results

Across all evaluations, Skip-Gram embeddings consistently outperform the TF-IDF baseline. Both the training objective and subsampling substantially affect semantic quality, embedding geometry, and computational efficiency.

A. Semantic quality: similarity and analogical structure

On WordSim-353, Skip-Gram achieves more than twice the correlation of TF-IDF with human similarity judgments. The strongest configuration is SG + NS with subsampling (Spearman $\rho = 0.4358$) compared to TF-IDF ($\rho = 0.201$). Subsampling improves performance for both objectives, with larger gains under Negative Sampling, indicating that reducing high-frequency noise sharpens semantic representations.

TABLE I: WordSim-353 Results (Spearman ρ)

Model	Sub	ρ	Used	OOV
SG + HS	No	0.3936	332	21
SG + HS	Yes	0.4282	332	21
SG + NS	No	0.3758	332	21
SG + NS	Yes	**0.4358**	332	21
TF–IDF	–	0.2010	332	21

On Google word analogies, TF-IDF performs near chance ($\approx 0.9\%$), while all Skip-Gram variants perform substantially better. SG + NS with subsampling again performs best (14.58%), demonstrating that Skip-Gram captures not only local similarity but also linear relational structure.

TABLE II: Google Analogy Task Accuracy

Model	Sub	Acc. (%)	Correct	Eval	OOV
SG + HS	No	13.66	1585	11602	7942
SG + HS	Yes	13.55	1572	11602	7942
SG + NS	No	11.14	1292	11602	7942
SG + NS	Yes	**14.58**	1692	11602	7942
TF–IDF	–	0.90	104	11602	7942

Frequency-binned analysis clarifies where subsampling helps most. For WordSim, subsampling yields the largest gains for low-frequency words, especially under Negative Sampling (low-frequency ρ increases from $\approx$ 0.38 to 0.57). For analogies, low-frequency questions remain difficult for all models ($\approx$ 6–7%), but subsampling strongly improves medium- and high-frequency accuracy for NS. These results indicate an interaction effect: Negative Sampling benefits disproportionately when frequent-word noise is suppressed.

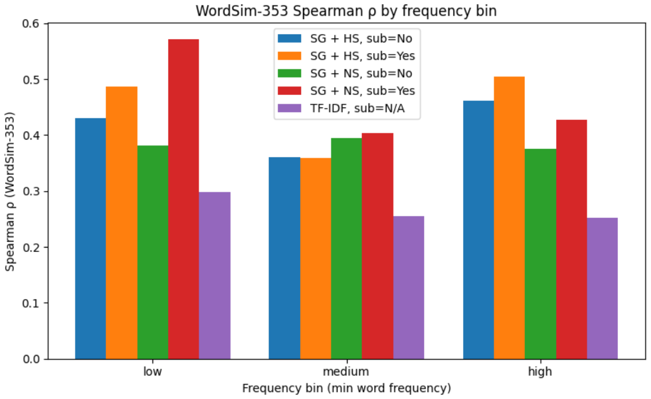

Fig. 1: Performance by frequency bin. (a) WordSim-353 Spearman ρ by frequency bin.

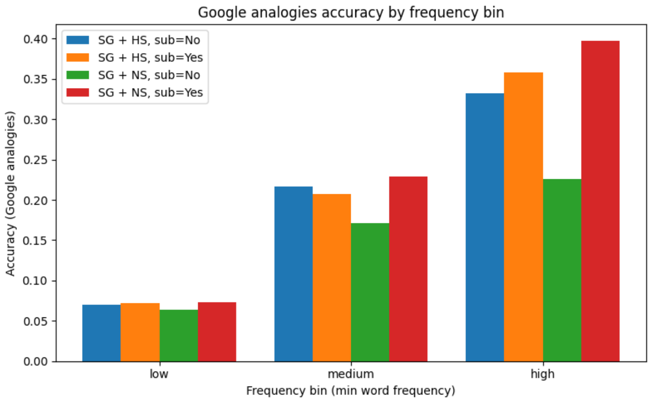

Fig. 1: Performance by frequency bin. (b) Google analogies accuracy by frequency bin.

Qualitative nearest-neighbor inspection is consistent with these findings: Skip-Gram neighborhoods are semantically coherent, while TF-IDF neighbors often reflect surface-level co-occurrence rather than meaning.

B. Global structure: clustering and low-dimensional organization

Clustering evaluations further confirm the superiority of Skip-Gram embeddings. When clustering words into five semantic categories, Skip-Gram achieves strong alignment with ground-truth labels (NMI = 0.9157, ARI = 0.8507), while TF-IDF performs poorly (NMI = 0.3817, ARI = 0.1576).

TABLE III: Clustering Performance (5 Semantic Categories)

Model	Sub	NMI	ARI
SG + NS	Yes	0.9157	0.8507
SG + HS	Yes	0.9157	0.8507
TF–IDF	–	0.3817	0.1576

This shows that Skip-Gram learns coherent global semantic structure, not just local neighborhoods.

Low-dimensional PCA and t-SNE visualizations reinforce this result: Skip-Gram embeddings form well-separated semantic regions, most clearly under Negative Sampling with subsampling, whereas TF-IDF exhibits no stable semantic organization.

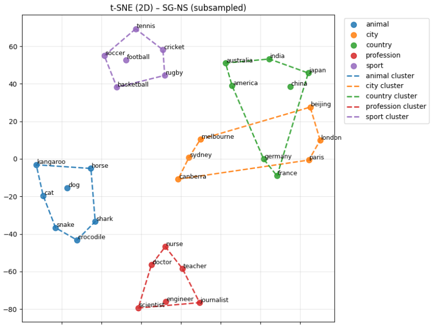

Fig. 2: Low-dimensional visualization of clustered words under Skip-Gram.

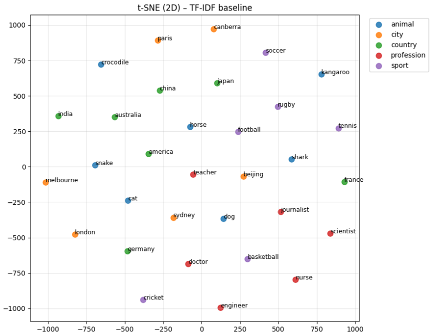

Fig. 3: Low-dimensional visualization of clustered words under TF–IDF.

C. Embedding-space geometry: frequency effects and neighborhood health

To connect performance to geometry, we analyze the relationship between embedding norm and word frequency. All Skip-Gram variants show a negative dependence, but the strength varies by objective and subsampling. The strongest coupling occurs for SG + NS without subsampling (slope -0.183, $R^2 = 0.646$), indicating frequency-driven distortion. With subsampling, this dependence flattens substantially (slope -0.0317, $R^2 = 0.2648$), demonstrating that subsampling acts as geometric regularization. In contrast, TF-IDF shows a strong positive coupling (slope $+6.271$, $R^2 = 0.839$), confirming that vector magnitudes are dominated by frequency rather than semantics.

TABLE IV: Linear regression: embedding norm vs. $\log_{10}(\text{freq})$

Model	Sub	Slope	Intercept	R^2
SG + HS	No	-0.0352	2.0517	0.2142
SG + HS	Yes	-0.0654	2.0954	0.3051
SG + NS	No	-0.1830	2.2509	0.6460
SG + NS	Yes	-0.0317	2.0473	0.2648
TF–IDF	–	$+6.2711$	-6.1008	0.8386

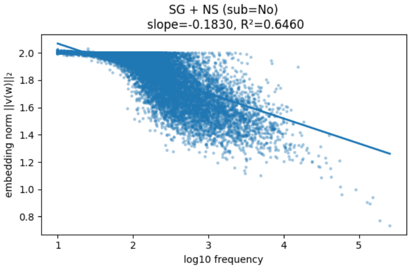

Fig. 4: Embedding norm vs. log-frequency for SG+NS without subsampling.

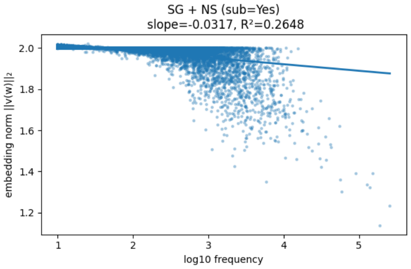

Fig. 4: Embedding norm vs. log-frequency for SG+NS with subsampling.

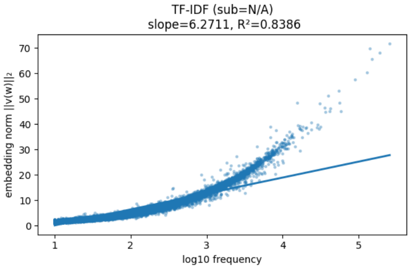

Fig. 4: Embedding norm vs. log-frequency for TF–IDF.

Hubness analysis further distinguishes the models. TF-IDF exhibits severe hub dominance (maximum hub count 89, P95 14), while Skip-Gram embeddings are far more balanced. Negative Sampling produces the healthiest neighborhood structure overall, achieving the lowest maximum hub concentration among all models. Subsampling has a more modest and mixed effect on hubness, slightly redistributing neighbor counts without uniformly reducing extreme hubs across objectives.

TABLE V: Hubness statistics across models ($k = 10$, ≈10,000 queries)

Model	Sub	Max hub	P95 hub
SG + HS	Yes	57	10
SG + HS	No	50	10
SG + NS	Yes	45	9
SG + NS	No	**35**	**9**
TF–IDF	–	89	14

D. Computational efficiency: time, throughput, and resource cost

Efficiency results reveal a clear cost–quality tradeoff. Negative Sampling achieves an order-of-magnitude higher throughput than Hierarchical Softmax (≈ 2000–2750 vs. 260–280 pairs/s) while reaching comparable or better semantic quality. Subsampling dramatically reduces wall-clock time, allowing strong performance to be achieved at much lower computational cost. Non-subsampled Hierarchical Softmax occupies the highest-cost regime without accuracy benefits.

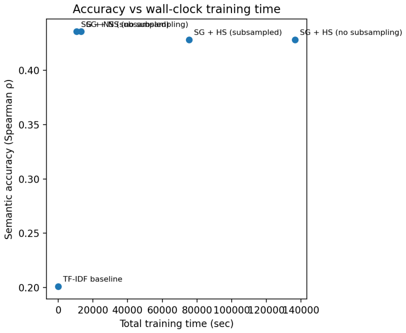

Fig. 5: Semantic accuracy versus total training time across models.

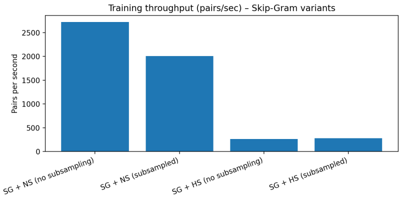

Fig. 5: Training throughput (pairs per second) for Skip-Gram variants.

Resource monitoring shows that training is CPU-bound across all Skip-Gram configurations. Subsampling primarily reduces total runtime and peak memory rather than instantaneous CPU load. Peak RAM drops substantially under subsampling, especially for HS (≈ 4.0 GB → 2.5 GB) and also for NS (≈ 4.1 GB → 2.25 GB). TF-IDF is computationally cheap, but its semantic performance is consistently inferior.

TABLE VI: Computational efficiency and resource cost

Model	Pairs/s	Time (s)	RAM (MB)
SG + NS (no sub.)	2726	13139	4053
SG + NS (sub.)	2007	10493	2239
SG + HS (no sub.)	262	136497	4031
SG + HS (sub.)	279	75527	2505
TF–IDF	686081	13	1431

Overall, the results support three conclusions: Skip-Gram embeddings strongly outperform TF-IDF on semantic tasks, Negative Sampling with subsampling is the best overall configuration, and subsampling serves not only as an efficiency mechanism but also as a form of geometric regularization that improves representation quality.

VI. Discussion: Interpretation, Tradeoffs, and Limitations

Taken together, the results show that Skip-Gram learns representations that reflect semantic structure rather than surface frequency patterns. Across evaluations, predictive training consistently produces embeddings with meaningful similarity, relational structure, and healthier geometry, while TF-IDF remains dominated by lexical overlap and frequency effects. This contrast highlights the difference between representation learning and purely statistical weighting.

An important observation is the role of subsampling. Rather than acting only as a computational shortcut, subsampling improves the quality and stability of the learned representations, especially under Negative Sampling. By reducing the influence of extremely frequent words, subsampling reshapes the effective learning problem and leads to embeddings that are both more efficient to train and more semantically coherent.

The comparison between training objectives reveals a clear practical tradeoff. Negative Sampling provides substantial efficiency gains and, when combined with subsampling, delivers the strongest overall performance. Hierarchical Softmax is more computationally expensive and less competitive in accuracy, though it appears somewhat less sensitive to the absence of subsampling. These differences reflect inherent algorithmic tradeoffs between contrastive learning and tree-based normalization.

There are also clear limitations. The analysis focuses on intrinsic evaluations and does not measure downstream task performance. Analogy accuracy remains modest, particularly for rare words, and results depend on specific choices of corpus, vocabulary thresholds, and hyperparameters. Despite these limitations, the findings consistently support Skip-Gram, and especially Negative Sampling with subsampling, as a strong and practical approach to learning semantic word representations.

VII. Conclusion

This project provides an empirical comparison of Skip-Gram word embeddings and a TF-IDF baseline, showing that predictive embedding learning produces representations with substantially stronger semantic structure. Across similarity, analogical reasoning, clustering, and geometric analyses, Skip-Gram consistently outperforms TF-IDF, with the strongest overall results achieved by Negative Sampling combined with subsampling. These findings highlight how training objective and frequency-based regularization jointly shape both performance and representation geometry.

Beyond the specific results, this study reinforces the broader significance of Skip-Gram as a foundational method in modern NLP. By framing word meaning as a predictive learning problem, Skip-Gram marked a shift away from purely count-based representations toward learned semantic spaces. Later models such as GloVe and contextual representations like BERT build on this same principle, extending it with global objectives or deep contextualization. In this sense, the empirical patterns observed here reflect a broader trajectory in NLP, where representation learning and training objectives play a central role in capturing linguistic meaning.

Bibliography

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[3] L. Finkelstein et al., "Placing Search in Context: The Concept Revisited," in *Proc. 10th Int. Conf. on World Wide Web (WWW)*, 2001. (Introduces WordSim-353.)

[4] E. Gabrilovich, "WordSimilarity-353 Test Collection" (dataset page).

[5] ACL Wiki, "Google analogy test set (State of the art)."

[6] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[7] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988. (TF-IDF weighting family.)

[8] S. Robertson, "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF," *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[9] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901. (PCA foundation.)

[10] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[11] L. Hubert and P. Arabie, "Comparing Partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985. (Adjusted Rand Index.)

[12] N. X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010. (NMI + clustering comparison theory.)

[13] Rohit Kulkarni ("therohk"), "A Million News Headlines" (dataset, Kaggle).

[14] R. Kulkarni, "A Million News Headlines," Harvard Dataverse, 2018, doi:10.7910/DVN/SYBGZL.

[15] "Plain text Wikipedia (SimpleEnglish)" (dataset, Kaggle).

[16] Wikimedia Foundation, "Wikimedia Downloads / Dumps" (Wikipedia dump distribution).

[17] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proc. EMNLP*, 2014.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019.

[19] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.