A SIMPLE BUT TOUGH-TO-BEAT BASELINE FOR SENTENCE EMBEDDINGS

PRESENTED BY - YASHWANTH, PRAHAS, HARSHITH, YASHASWINI

ABSTRACT

The paper introduces a simple yet powerful unsupervised method for generating sentence embeddings.
Instead of relying on complex neural networks or large labeled datasets, the authors propose:

- Computing a weighted average of pre-trained word embeddings.
- Removing the first principal component (PCA/SVD) from the resulting sentence vectors.

This approach, though extremely simple, achieves performance comparable to — and often better than — supervised models such as RNNs and LSTMs on several semantic textual similarity tasks.

Key Concept: Simplicity, when combined with theoretical grounding, can outperform complex deep models.

INTRODUCTION

Background:

- Word embeddings (Word2Vec, GloVe) represent word meanings effectively.
- The challenge is to extend these representations to entire sentences that capture semantic meaning.

Why Do We Need Them?

- Traditional word embeddings (like Word2Vec or GloVe) only capture the meaning of individual words, not whole sentences.

- To understand full sentences, we need to combine word meanings in a way that captures context, importance, and sentence structure.

Objective of the paper:

- To develop an unsupervised, theoretically justified, and computationally simple baseline that can rival sophisticated neural models.

PROBLEM AND PREVIOUS METHODS

RNN: Read each word step by step.
Drawback: Accurate but slow, hard to train, needs labeled data.

Skip-Thought (Kiros et al., 2015) : Predicts the next sentence from the current one (like Word2Vec for sentences).
Drawback: Very large model, takes weeks to train.

Doc2Vec (Le & Mikolov, 2014) : Learns a unique vector for each sentence or document.
Drawback : Needs training on a specific corpus; not easily generalizable.
These models are computationally heavy ,supervised and often dont generalize well

Gaps:
Even simple averaging of word vectors performs surprisingly well.

This motivates investigating why and how such simplicity works — and improving it systematically.

THEORY

RANDOM WALK MODEL (ARORA, 2016)

$$\Pr[w \text{ emitted at time } t \mid c_t] \propto \exp\left(\langle c_t, v_w \rangle\right).$$

Limitations:

1. Some words appear out of context
E.g., “actually”, “however”, “the” sometimes show up even when they’re not semantically related.

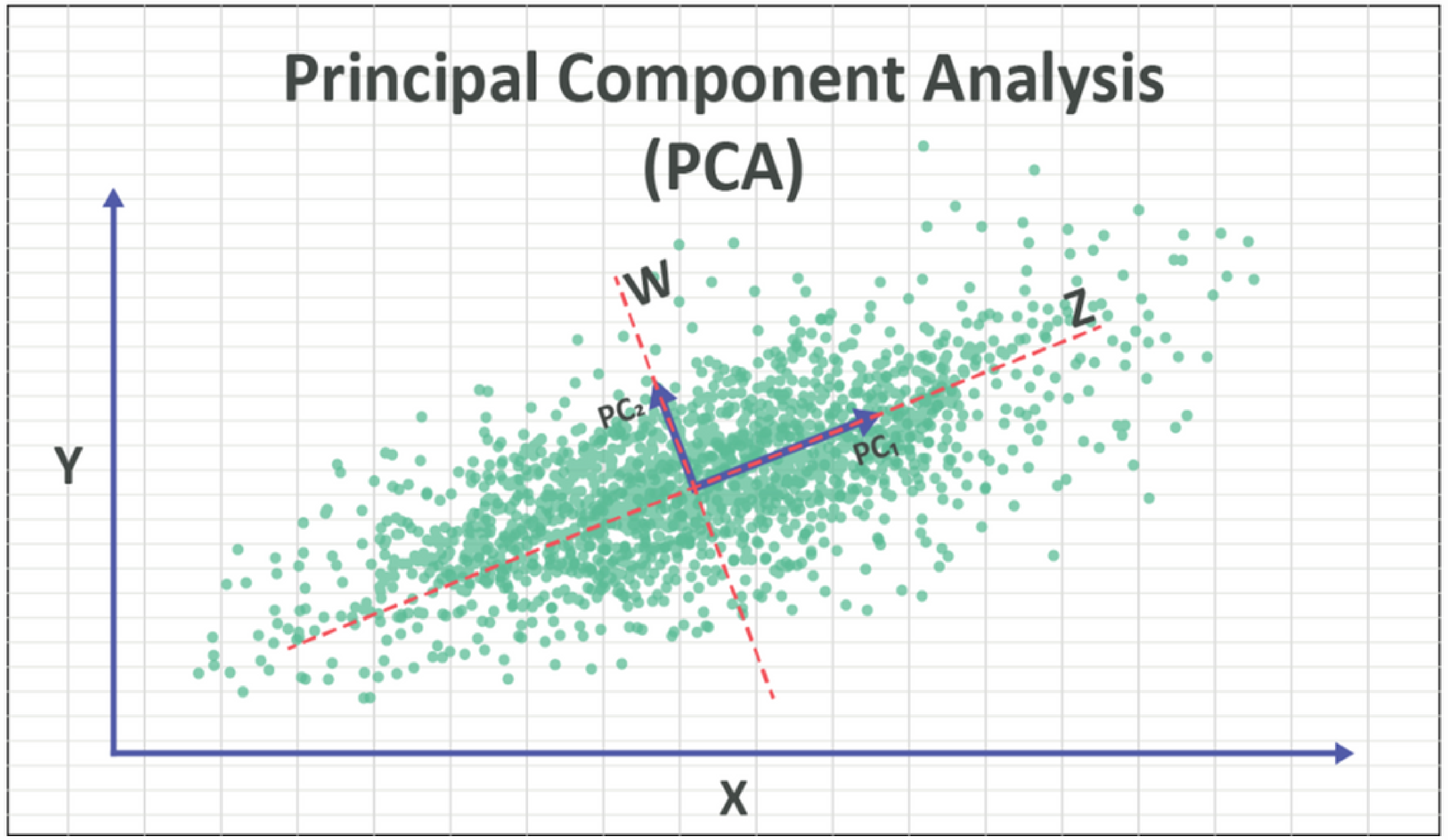
2. Some frequent words appear everywhere
E.g., Words like “the”, “and”, “of” appear across almost all topics.

IMPROVED RANDOM WALK MODEL

Steps to compute a sentence embedding:

1. Compute the weighted average of all word vectors in the sentence.
2. Perform PCA/SVD across all sentence vectors.
3. Subtract the projection on the first principal component (common direction).

$$\mathrm{Weight}(w) = \frac{a}{a + p(w)}$$

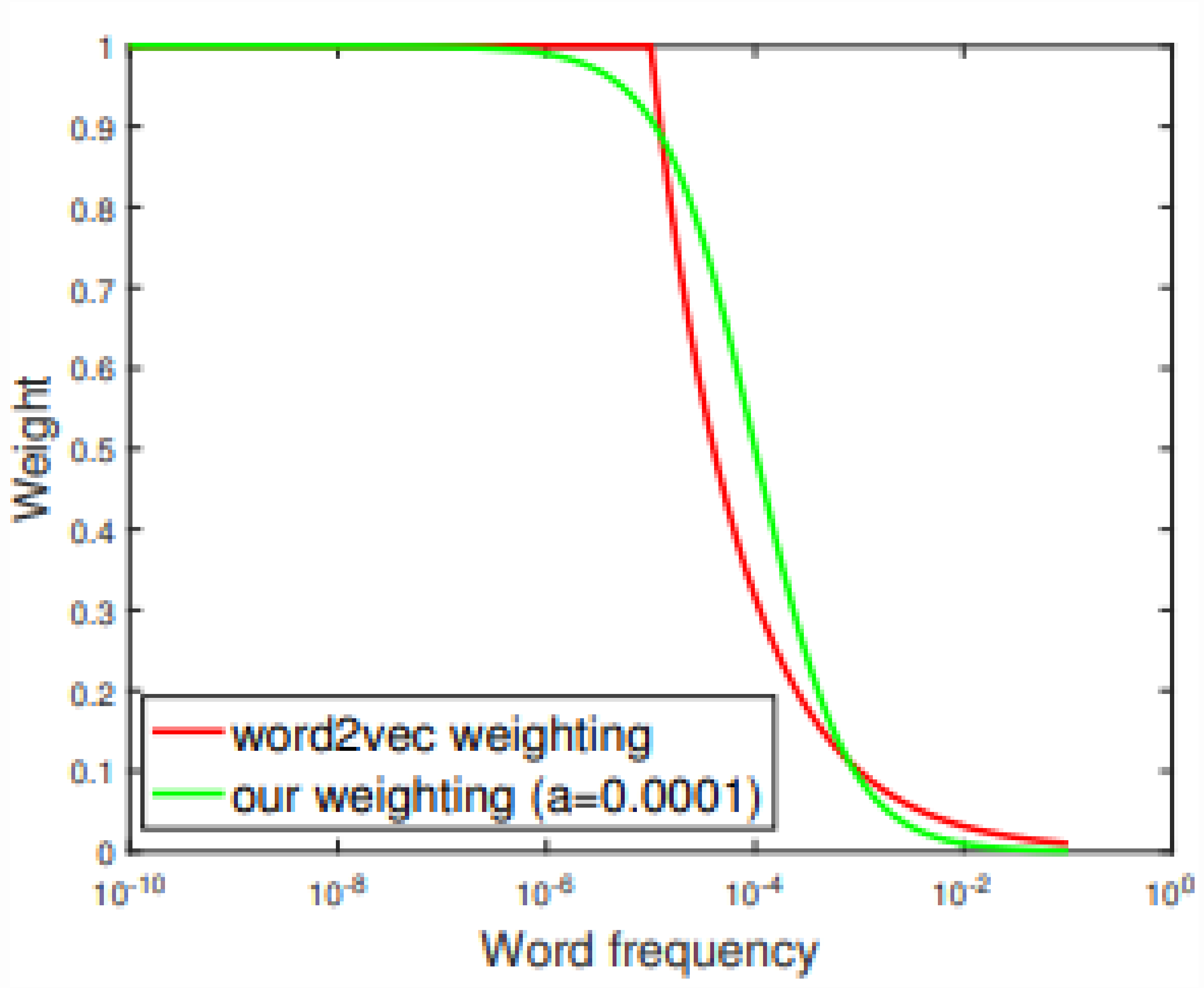


Concretely, given the discourse vector c_s, the probability of a word w is emitted in the sentence s is modeled by,

$$\Pr[w \text{ emitted in sentence } s \mid c_s] = \alpha p(w) + (1-\alpha)\frac{\exp\left(\langle \tilde{c}_s, v_w \rangle\right)}{Z_{\tilde{c}_s}}, \tag{2}$$

$$\text{where } \tilde{c}_s = \beta c_0 + (1-\beta)c_s, \;\; c_0 \perp c_s$$

$$\tilde{c}_s \propto \sum_{w \in s} \frac{a}{p(w)+a} v_w$$



The graph demonstrates that Word2Vec’s subsampling probabilities and Arora et al.’s theoretical weighting scheme behave similarly — both effectively downweight frequent, less informative words and emphasize rare, meaningful words when learning embeddings.

EXPERIMENTS

Datasets Used:

- 22 Semantic Textual Similarity (STS) datasets (SemEval 2012–2015).
- SICK 2014 (Semantic relatedness and entailment).
- Twitter Semantic Similarity dataset (2015).

Compared Methods:

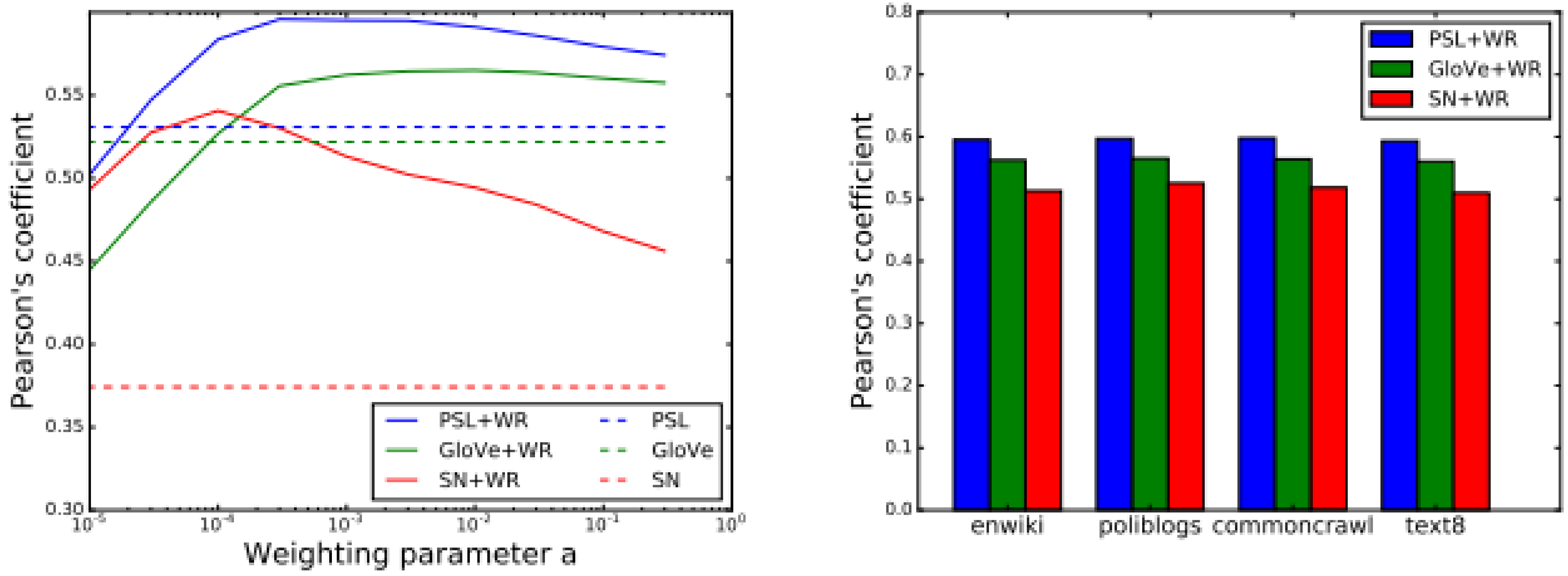
- Unsupervised: avg-GloVe, tfidf-GloVe, Skip-Thought.
- Semi-supervised: avg-PSL (trained on PPDB).
- Supervised: RNN, LSTM, DAN, and PP-proj models.

RESULTS SUMMARY

- GloVe+WR (Weighted + PCA Removal):
  - Improves by 10–30% over simple averaging.
  - Outperforms RNNs and LSTMs on many STS tasks.
- PSL+WR:
  - Best results overall (semi-supervised setting).
- Robust to:
  - Worked well on different datasets (Wikipedia, Blogs, Common Crawl).
  - Performance stayed stable for the parameter a in the range 10^{-3} to 10^{-4}.

Observation:
The method is simple, robust, and transferable across domains.

FIGURE 2: EFFECT OF WEIGHTING SCHEME IN OUR METHOD ON THE AVERAGE PERFORMANCE ON STS 2012 TASKS. BEST VIEWED IN COLOR



(a) (b)

CONCLUSION

- SIF (Smooth Inverse Frequency) is a simple and unsupervised method for generating sentence embeddings.
- Combines a weighted average of word vectors with common component removal (PCA).
- Performs better than unweighted averages and even beats or matches supervised RNN/LSTM models.
- Works exceptionally well in unsupervised settings, and performs even better when used with semi-supervised word embeddings (PSL + WR).
- Shuffled-word experiment:
    - SIF ignores word order but still performance well.
    - Shows that semantic meaning matters more in NLP tasks.
- Even a simple unsupervised method can match or outperform deep, complex neural models.

IMPACT SINCE 2017

- Became a benchmark baseline for all later sentence embedding methods.
- Inspired newer models like Universal Sentence Encoder (2018) and Sentence-BERT (2019).
- SIF remains widely used for fast, unsupervised text similarity.
- Still applied in retrieval, clustering, and domain adaptation tasks today.

VIELEN DANK