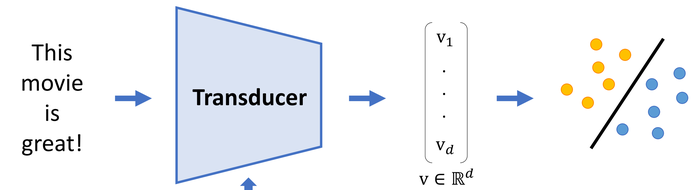
A Compressed Sensing View of Unsupervised Text Embeddings, Bag-of-n-Grams, and LSTMs

Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, Kiran Vodrahalli
{arora,mkhodak,nsaunshi}@cs.princeton.edu, kiran.vodrahalli@columbia.edu

A competitive document representation with provable properties

Modern NLP pipelines combine low-dimensional **distributed representations** of text with **deep learning** models like LSTMs. Our goal is to reason formally about these systems using compressed sensing tools.

The Text Classification Pipeline



Bag-of-n-Grams (BonGs)	Past Linear Schemes (e.g. sum of embeddings, SIF [1])	LSTM Hidden States (e.g. skip-thoughts [3])
Simple to implement	Simple to implement	Long range dependencies
Very strong baseline [4]	Low dimensional	Strong performance
High-dimensional	Moderate performance	Slow
Only local word-order	Often ignore word-order	**Can be beaten by BonGs**

Our Method

A simple linear scheme using word embeddings $\mathbf{v}_w \in \mathbb{R}^d$:

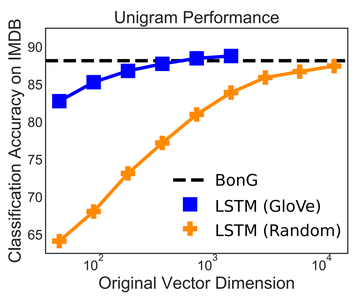
- ***n*-grams** $g = w_1, \ldots, w_n$ represented as element-wise products:

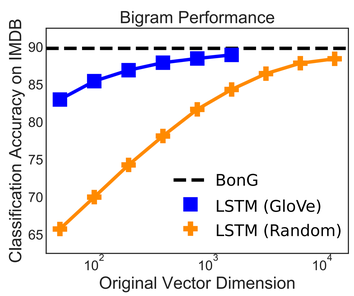
$$\mathbf{v}_g = \mathbf{v}_{w_1} \odot \cdots \odot \mathbf{v}_{w_n}$$

- **documents** represented as sums of their n-gram vectors:

$$\mathbf{v}_{\text{document}} = \sum_{g \in \text{ngrams}} \mathbf{v}_g$$

This representation is **provably as strong as Bag-of-n-Grams** on linear text classification, **can be computed by a low-memory LSTM**, and **performs well** on a variety of tasks in practice.

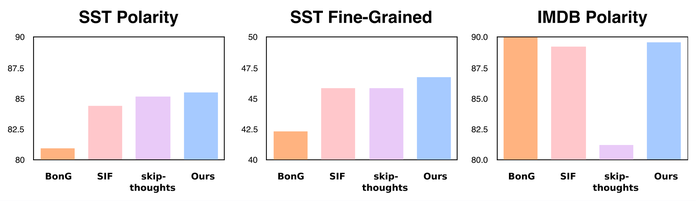




How well does our representation do on linear text classification?

Case 1: Random Word Embeddings:
Using i.i.d. Rademacher word embeddings as input our representations are *provably* as powerful as Bag-of-n-Grams for linear text classification. This yields a new theoretical result about LSTMs (below).

Case 2: Pretrained Word Embeddings:
Using GloVe word embeddings our representations achieve state-of-the-art results on several text classification tasks:



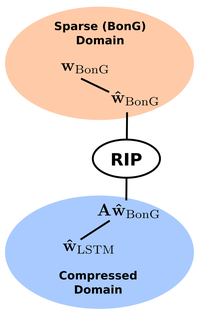
Theorem: LSTMs beat BonGs

If ℓ is a convex Lipschitz loss and $\mathcal{D}$ is a distribution on documents of length at most T with optimal linear BonG classifier $\mathbf{w}_{\text{BonG}}$ then for $d = \tilde{\Omega}\left(\frac{T}{\varepsilon^2}\log\frac{1}{\delta}\right)$ one can initialize an $\mathcal{O}(d)$-memory LSTM such that with probability $1-\delta$ the linear classifier $\hat{\mathbf{w}}_{\text{LSTM}}$ trained over m documents represented by the LSTM's last hidden state satisfies

$$\ell_{\mathcal{D}}(\hat{\mathbf{w}}_{\text{LSTM}}) \leq \ell_{\mathcal{D}}(\mathbf{w}_{\text{BonG}}) + \mathcal{O}\left(\|\mathbf{w}_{\text{BonG}}\|_2\sqrt{\varepsilon + \frac{1}{m}\log\frac{1}{\delta}}\right)$$

Proof Sketch: Using results from compressed sensing we can write $\mathbf{v}_{\text{document}} = \mathbf{A}\mathbf{v}_{\text{BonG}}$, where the matrix $\mathbf{A}$ preserves inner products of T-sparse vectors up to distortion ε and $\mathbf{v}_{\text{BonG}}$ is the document's BonG vector. As $\mathbf{v}_{\text{document}}$ can be computed by a low-memory LSTM, it suffices to show that **learning is possible under compression** [2]:

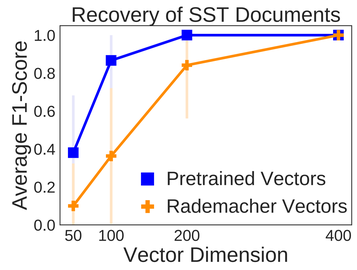
1. The loss of learned classifier $\hat{\mathbf{w}}_{\text{BonG}}$ is bounded in terms of that of the optimal classifier $\mathbf{w}_{\text{BonG}}$.
2. $\hat{\mathbf{w}}_{\text{BonG}}$ can be expressed as a linear combination of BonGs. Since $\mathbf{A}$ preserves their inner products and the loss is Lipschitz, the loss of $\mathbf{A}\hat{\mathbf{w}}_{\text{BonG}}$ is thus bounded in terms of that of $\hat{\mathbf{w}}_{\text{BonG}}$.
3. The loss of learned classifier $\hat{\mathbf{w}}_{\text{LSTM}}$ is bounded in terms of that of $\mathbf{A}\hat{\mathbf{w}}_{\text{BonG}}$.

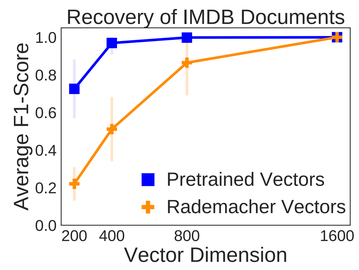


What information does our representation encode?

Case 1: Random Word Embeddings:
Guaranteed polynomial-time recovery of the Bag-of-n-Grams vector from our representation using ℓ_1-minimization. Follows from the compressed sensing properties of random matrices.

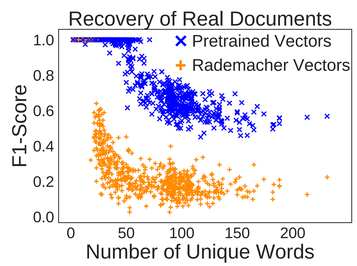
Case 2: Pretrained Word Embeddings:
Standard compressed sensing theory does not apply to GloVe/word2vec. Surprisingly, they encode Bag-of-Words vectors *more efficiently* than random embeddings, requiring fewer dimensions for recovery:

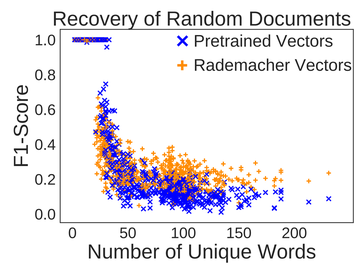




Empirical Observation

As a result of being trained on a large text corpus, word embeddings satisfy a weak compressed sensing condition that only holds for natural language documents. This leads to highly-efficient BoW recovery.





References

[1] Arora et al. *A Simple but Tough-to-Beat Baseline for Sentence Embeddings.* ICLR 2017.
[2] Calderbank et al. *Compressed Learning.* Technical Report 2009.
[3] Kiros et al. *Skip-Thought Vectors.* NIPS 2015.
[4] Wang & Manning. *Baselines and Bigrams.* ACL 2012.