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

Sanjeev Arora

Misha Khodak

Nikunj Saunshi

Kiran Vodrahalli

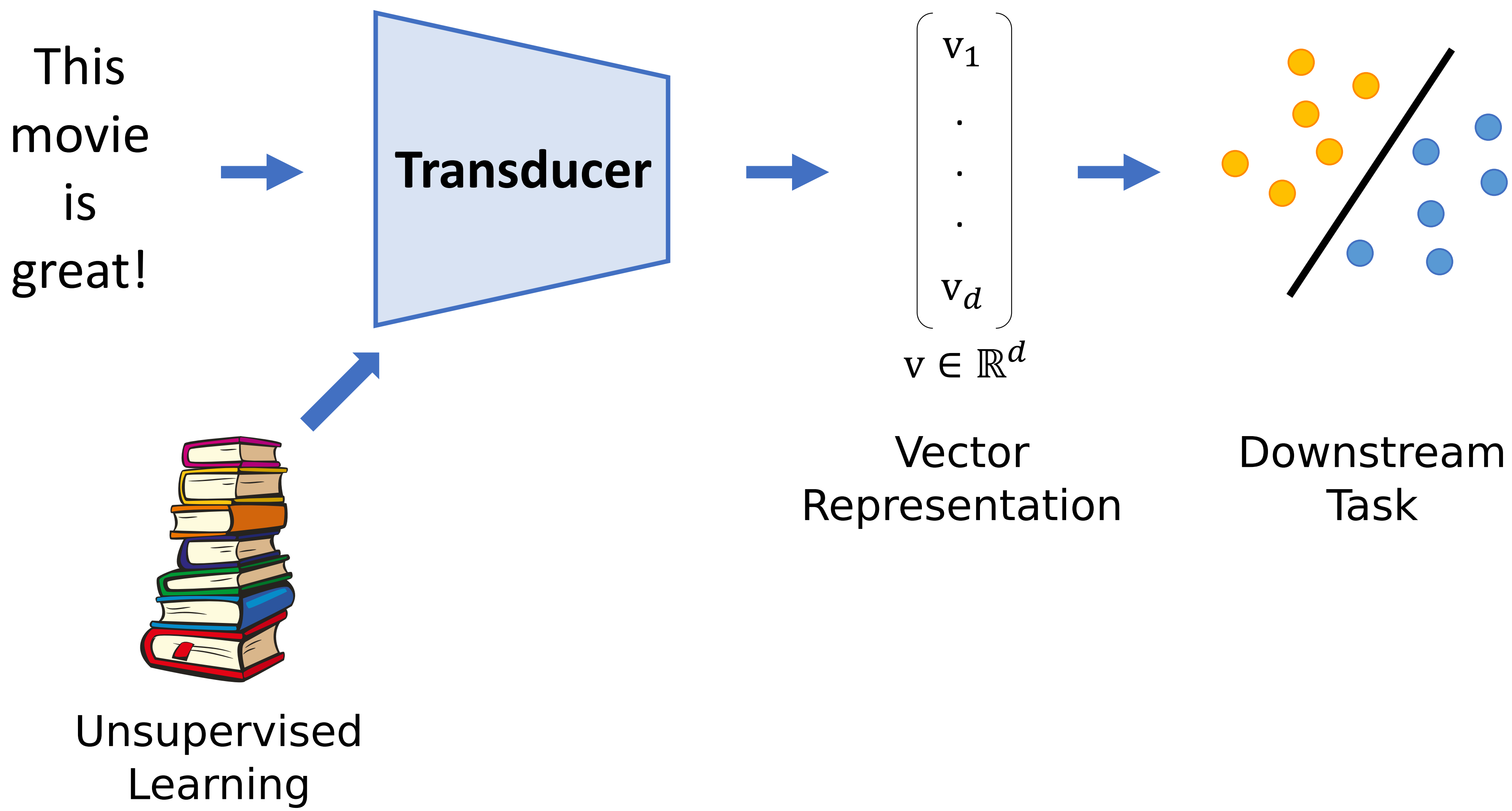
Modern unsupervised text embeddings

NLP practitioners use unsupervised text embeddings to capture the “meaning” of documents.

Often produced or taken as input by (recurrent) neural networks.

Goal: compete with this state-of-the-art using simple, analyzable, deep-learning free methods

Why represent text as an embedding?

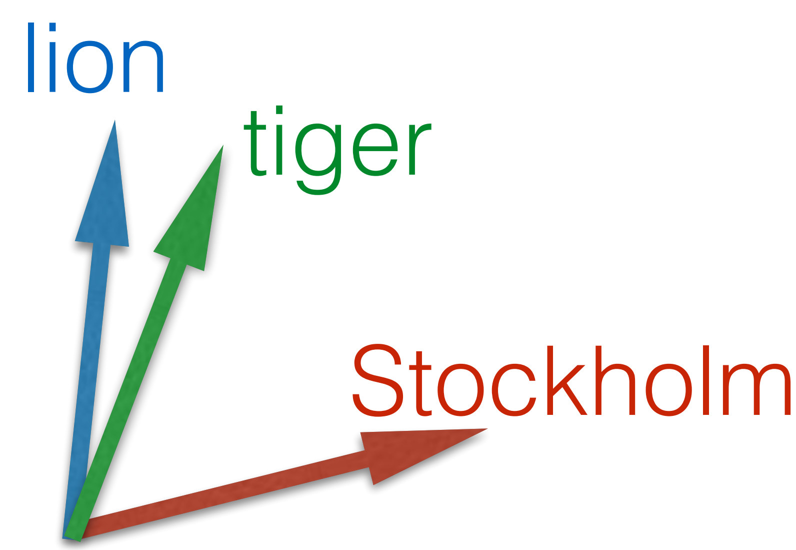


Want to use large amounts of unsupervised data to improve performance/sample efficiency on supervised tasks.

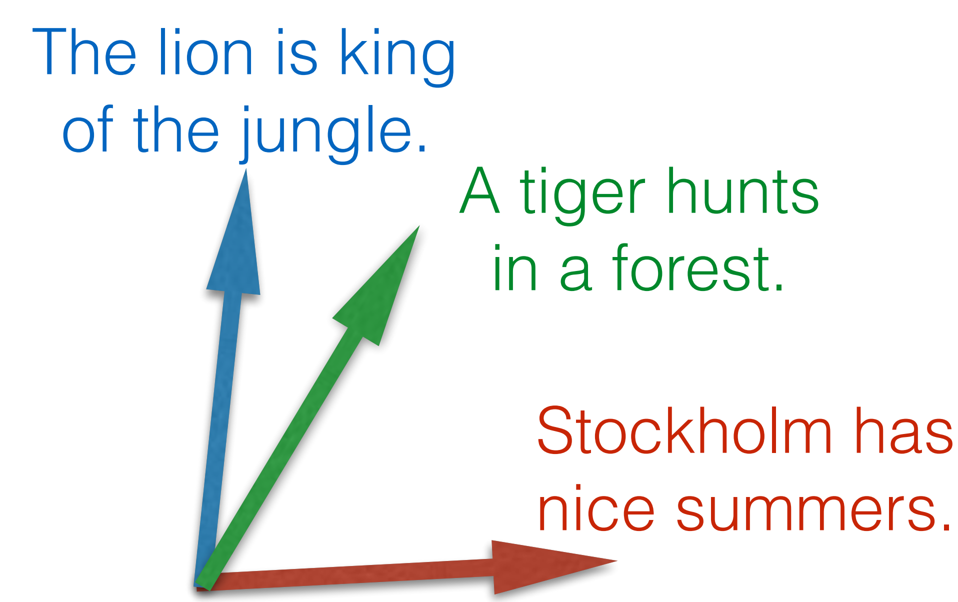
How to represent text as an embedding?

Word embeddings:

- Assign vector to each word (dimension d~300)
- optimize objective that makes frequently co-occurring words have high inner product (e.g. word2vec[1] or GloVe[2])



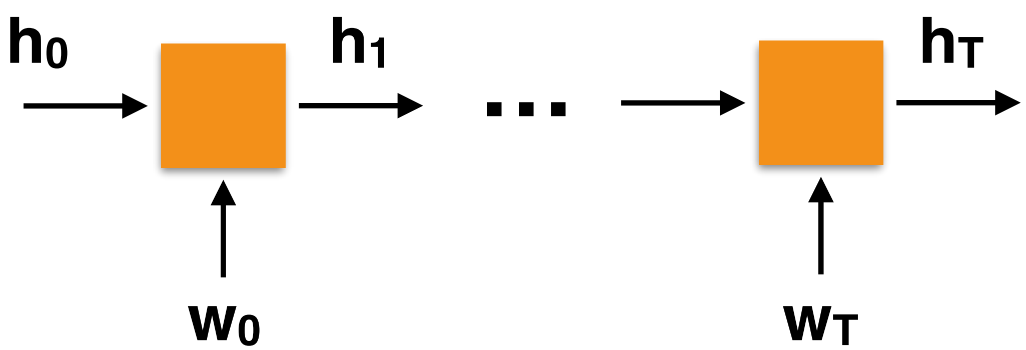
How to extend to longer text?



1: Mikolov et al., *NIPS* 2013.
2: Pennington et al., *EMNLP* 2014.

The LSTM embedding approach

Take in words $\mathbf{w_1, \ldots, w_T}$ and compute a hidden state vector $\mathbf{h_t}$ at each step. The embedding is the last state $\mathbf{h_T}$:



$$h_t = F(v_{w_t}, h_{t-1})$$

$$f(v_{w_t}, h_{t-1}) \circ h_{t-1} + i(v_{w_t}, h_{t-1}) \circ g(v_{w_t}, h_{t-1}))$$

Examples:

- skip-thought (Kiros et al., 2015)
- MC-QT (Logeswaran and Lee, 2018)

Drawbacks:

- slow for training and inference
- struggles against **Bag-of-n-Grams (BonG)** — sparse vectors counting the n-grams in a document — on text classification

Many attempts at simple embeddings

The embedding is a sum of word embeddings (perhaps weighted or linearly transformed):

$$v_{w_1,\ldots,w_T} = \sum_{i=1}^{T} v_{w_i}$$

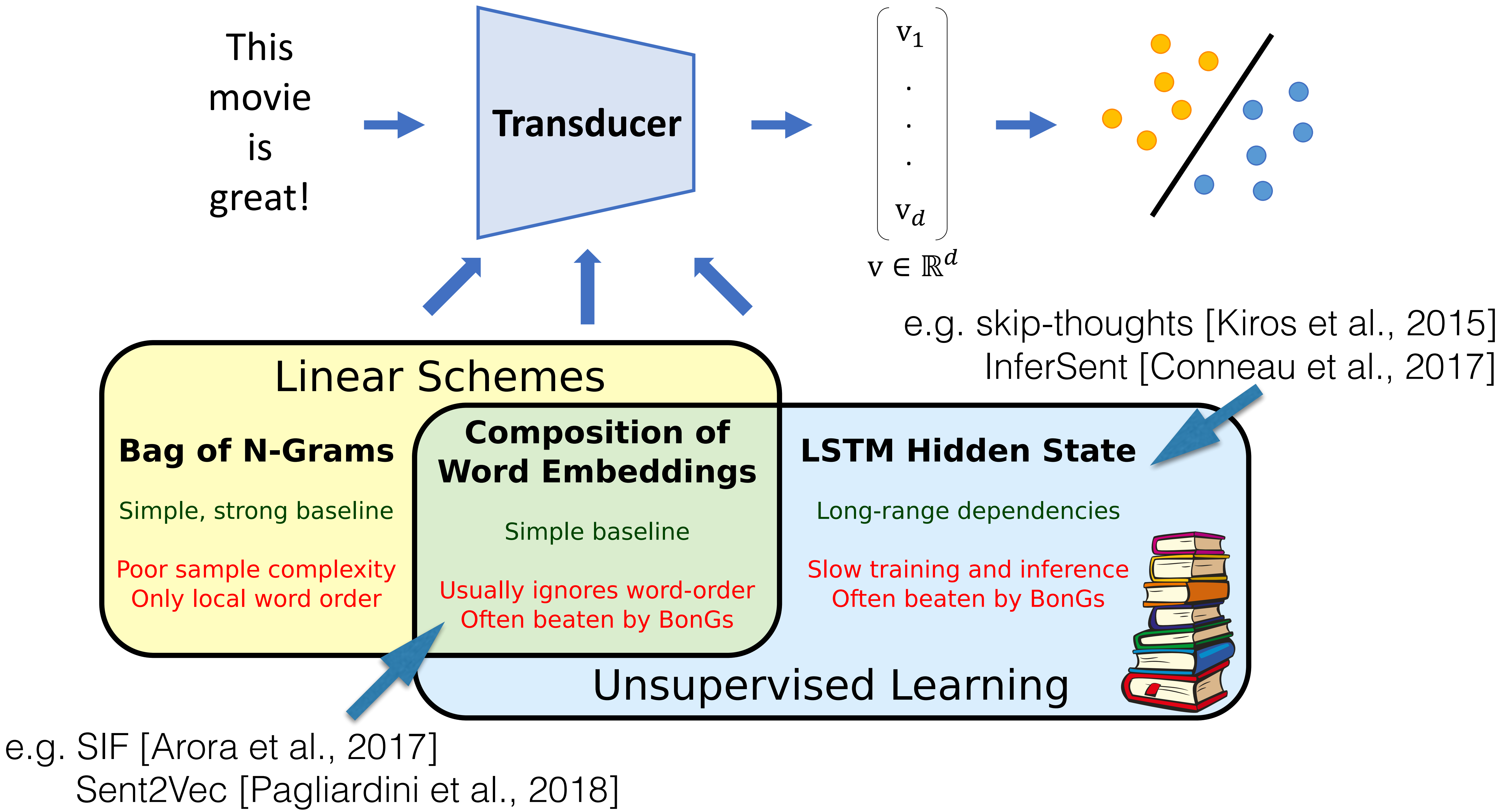
Examples:

- paraphrastic — use word vectors trained on a corpus of paraphrases (Wieting et al., 2016)
- SIF — down-weight frequent words (Arora et al., 2017)

Drawbacks:

- have not incorporated word-order information successfully
- not as successful on classification as on semantic similarity

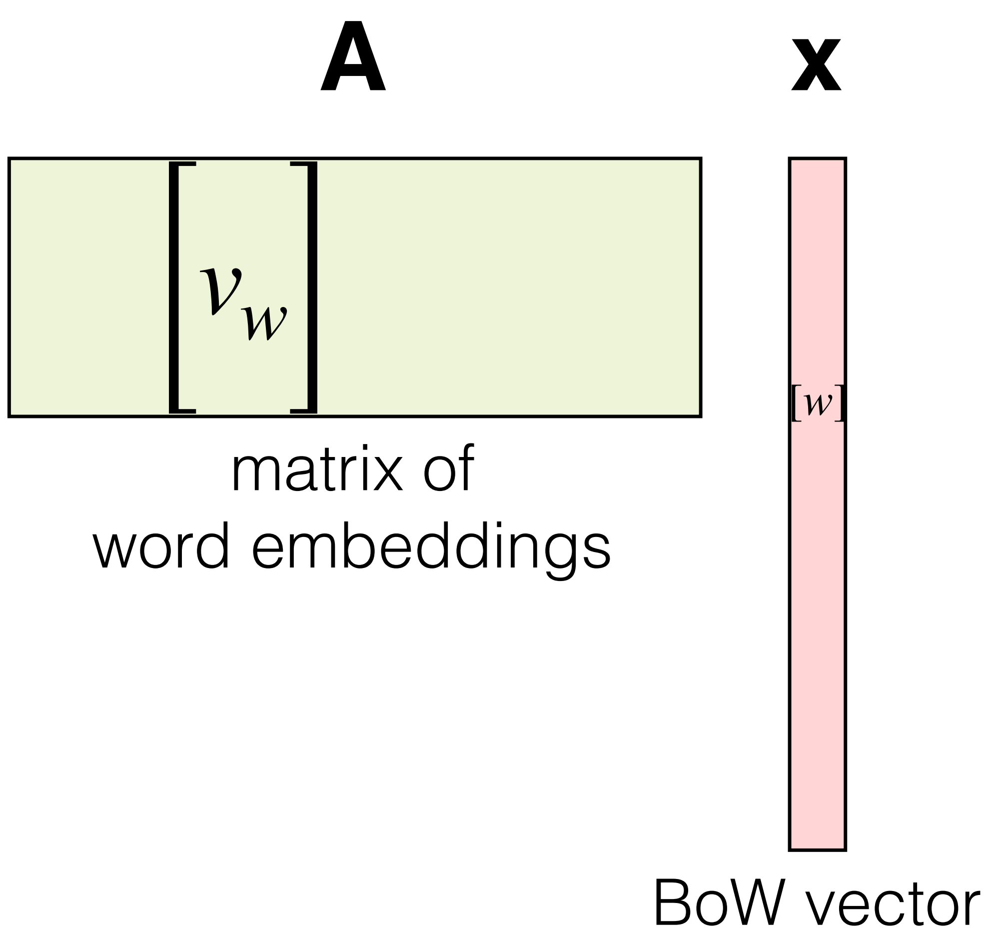
Summary of text embedding methods



What to aim for in the unsupervised setting

Task unknown beforehand — maybe try and preserve most of the information in the text in an easily extractable way?

This is what the sum-of-embeddings does with the Bag-of-Words vector:



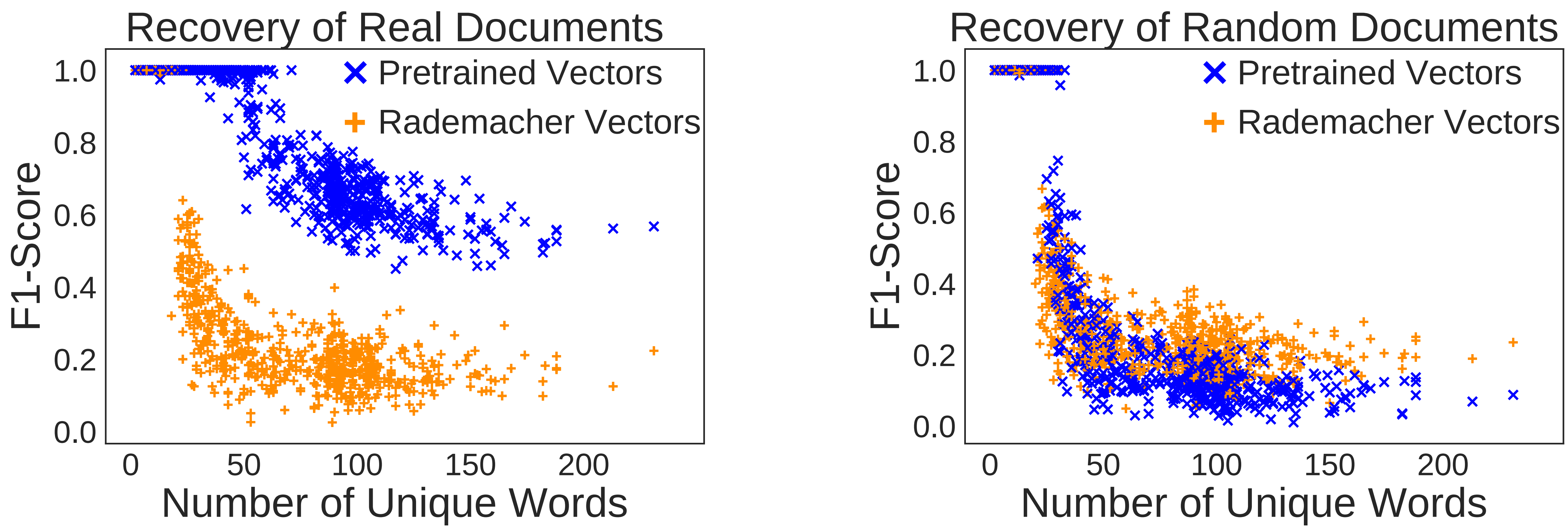
recovering *sparse* **x** given **Ax** is the goal of compressed sensing [Donoho, 2005; Candès & Tao, 2005]

can be done if the columns of **A** are incoherent or near-orthogonal

Can we recover information from a sum of pretrained embeddings?

Experiment:

1. compress $\mathbf{x}_{\mathbf{BoW}}$ as $\mathbf{A}\mathbf{x}_{\mathbf{BoW}}$
2. recover $\mathbf{x}_{\mathbf{BoW}}$ by Basis Pursuit: $\min \|x\|$ s.t. $Ax = Ax_{\mathrm{BoW}}$



Yes! We are more likely to recover $\mathbf{x}_{\mathbf{BoW}}$ from $\mathbf{A}\mathbf{x}_{\mathbf{BoW}}$ using Basis Pursuit if $\mathbf{A}$ consists of pretrained embeddings and $\mathbf{x}_{\mathbf{BoW}}$ comes from a *real document*.

Does recoverability imply learnability in the compressed domain?

Yes, for random vectors:

For **A** satisfying RIP, linear classification over compressed samples **Ax** is approximately at least as good as over **x**, *assuming x is sparse*.

Can construct such **A** w.h.p. using random vectors with dimension O(k log N / k).

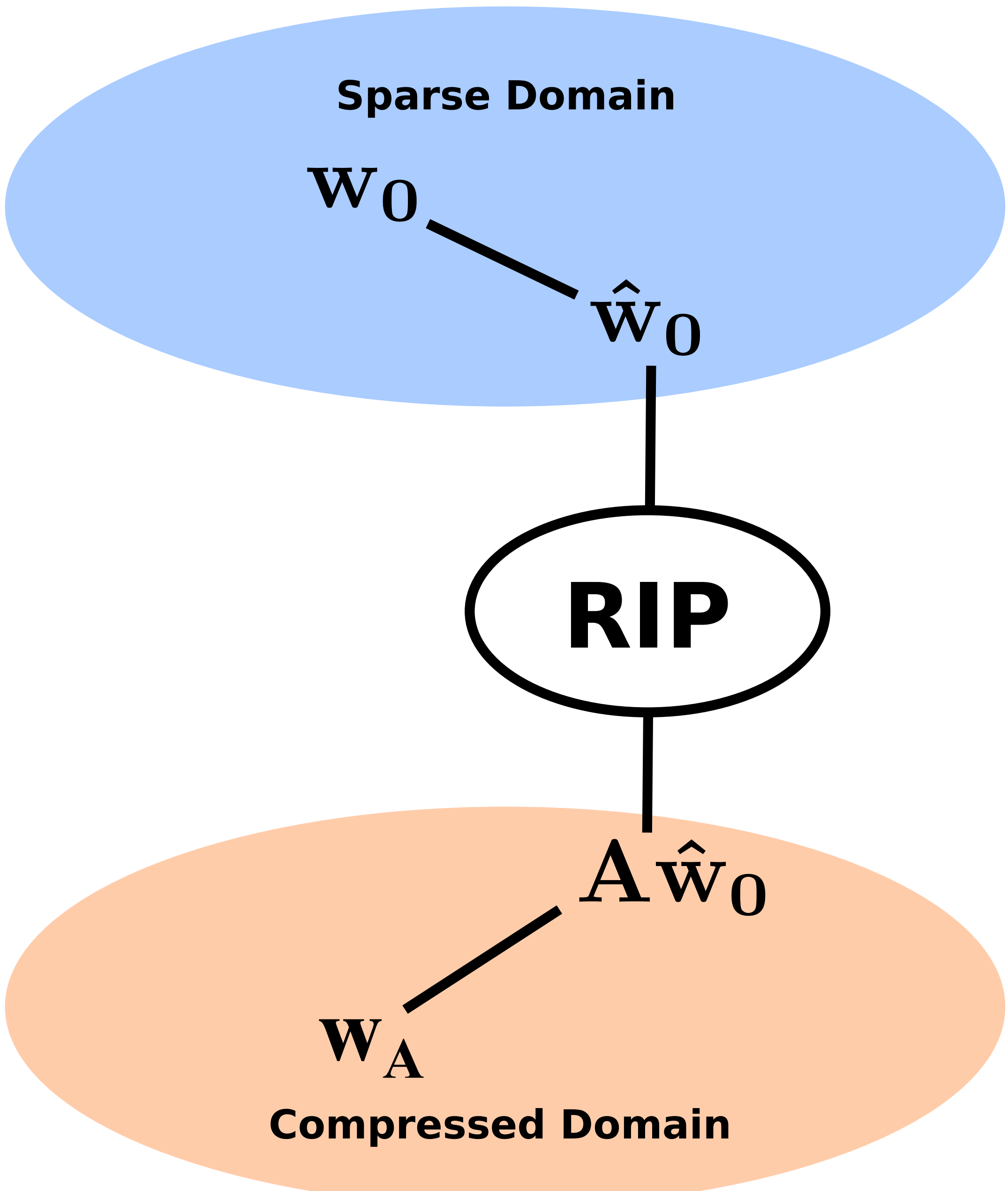
Restricted Isometry Property: $A \in \mathbb{R}^{d \times N}$ is (k, ε)-RIP if

$$(1-\varepsilon)\|x\| \leq \|Ax\| \leq (1+\varepsilon)\|x\| \ \forall\ k\text{-sparse } x \in \mathbb{R}^N$$

More formally

Theorem 1: If the distribution D of examples ($\mathbf{x}$, y) has **k-sparse** $\mathbf{x}$, $\mathbf{w_0}$ is their optimal linear classifier for some convex Lipschitz loss, and **A is (2k,ε)-RIP**, then the linear classifier $\mathbf{w_A}$ trained over ($\mathbf{Ax}$, y) satisfies:

$$\ell_D(w_A) \leq \ell_D(w_0) + O\left(\sqrt{\varepsilon}\right)$$



Proof Sketch:

classifier $\hat{w}_0$ is a linear combination of training examples

A is ε – RIP $\implies (A\hat{w}_0)^T Ax \leq \hat{w}_0^T x + O(\varepsilon)$

ℓ is Lipschitz $\implies \ell(A\hat{w}_0) \leq \ell(\hat{w}_0) + O(\varepsilon)$

4: Extends results by Calderbank et al. (*Technical Report* 2009).

Compressing Bag-of-n-Grams Information

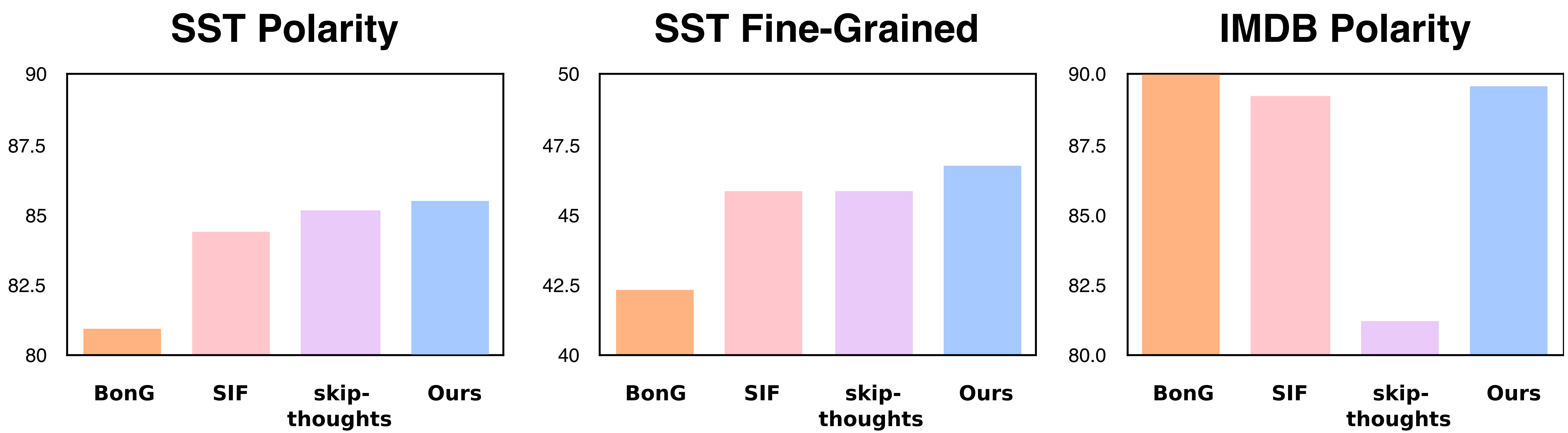
By Theorem 1 the sum of embeddings is as good as the Bag-of-Words for RIP vectors. **But we want to be as good as *Bag-of-n-Grams*.**

Our approach — take a sum over n-gram embeddings:

- For n-gram $g=(w_1,\ldots,w_n)$ set $v_g = v_{w_1} \odot \cdots \odot v_{w_n}$
- With some assumptions we can show these vectors satisfy RIP, so their sums are guaranteed to do as well as Bag-of-n-Grams.
- We call these ***DisC embeddings*** (for *distributed co-occurrence*).

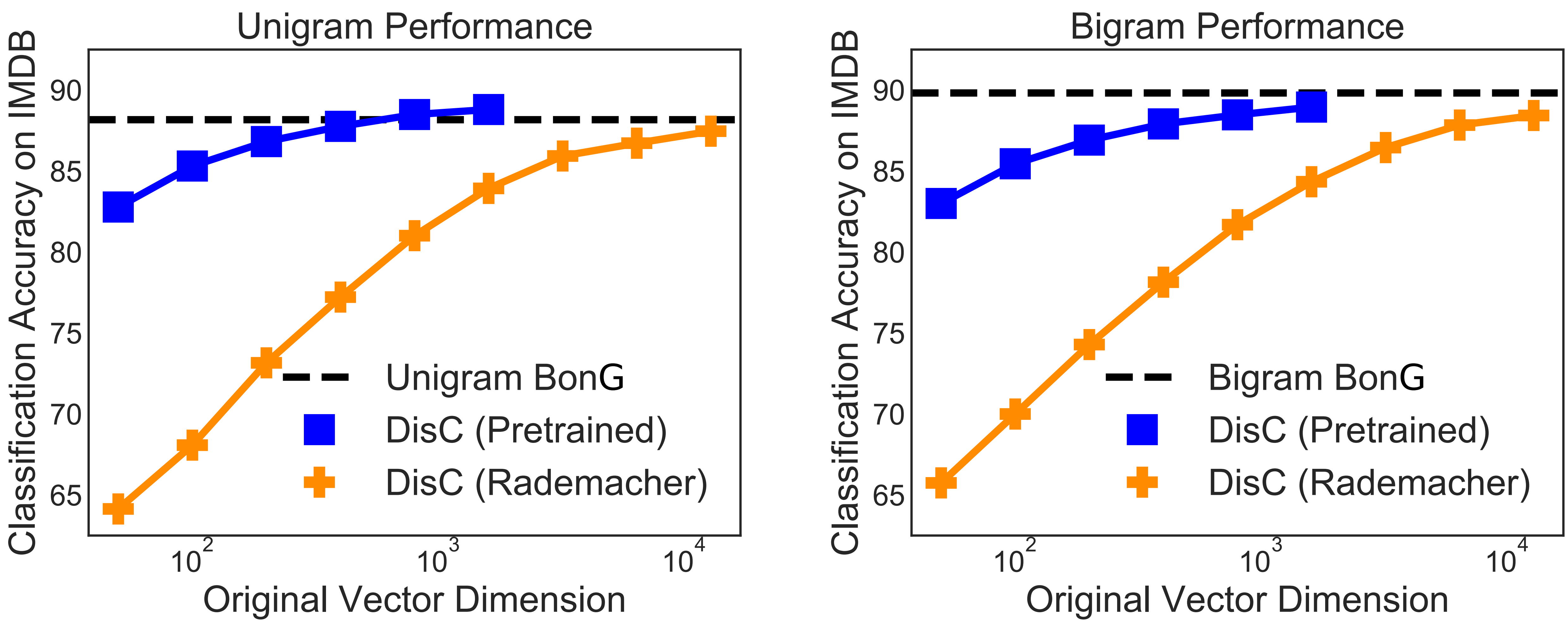
Properties of DisC embeddings

- perform well on standard classification tasks, competing with latest neural methods:



- can be constructed by a low-memory LSTM, so by Theorem 1 **even a linear LSTM can do at least as well as Bag-of-n-Grams** on text classification (if initialized properly)

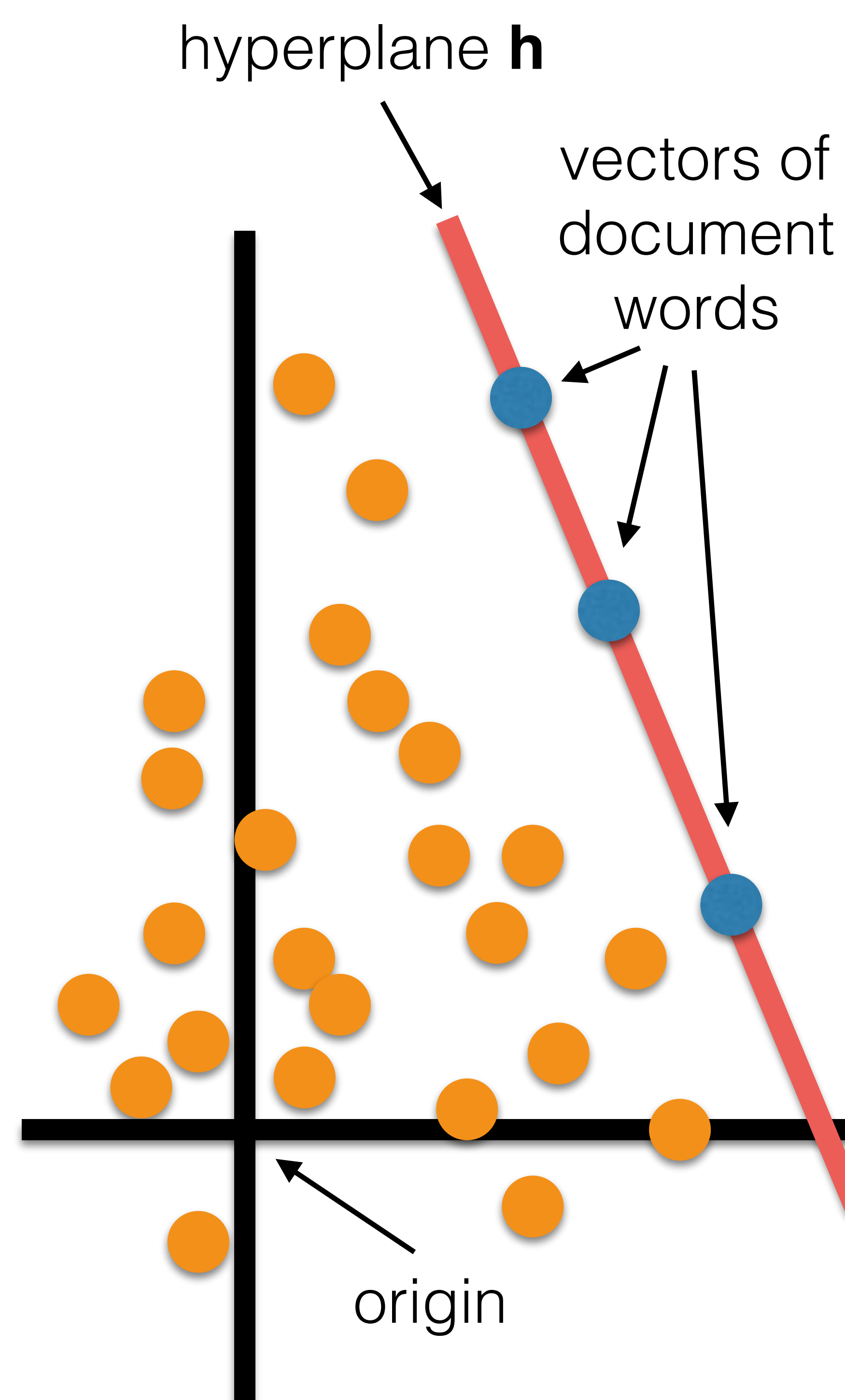
Verifying our theory: convergence to Bag-of-n-Grams performance



Using pretrained embeddings yields much better performance, even though they do not satisfy RIP.

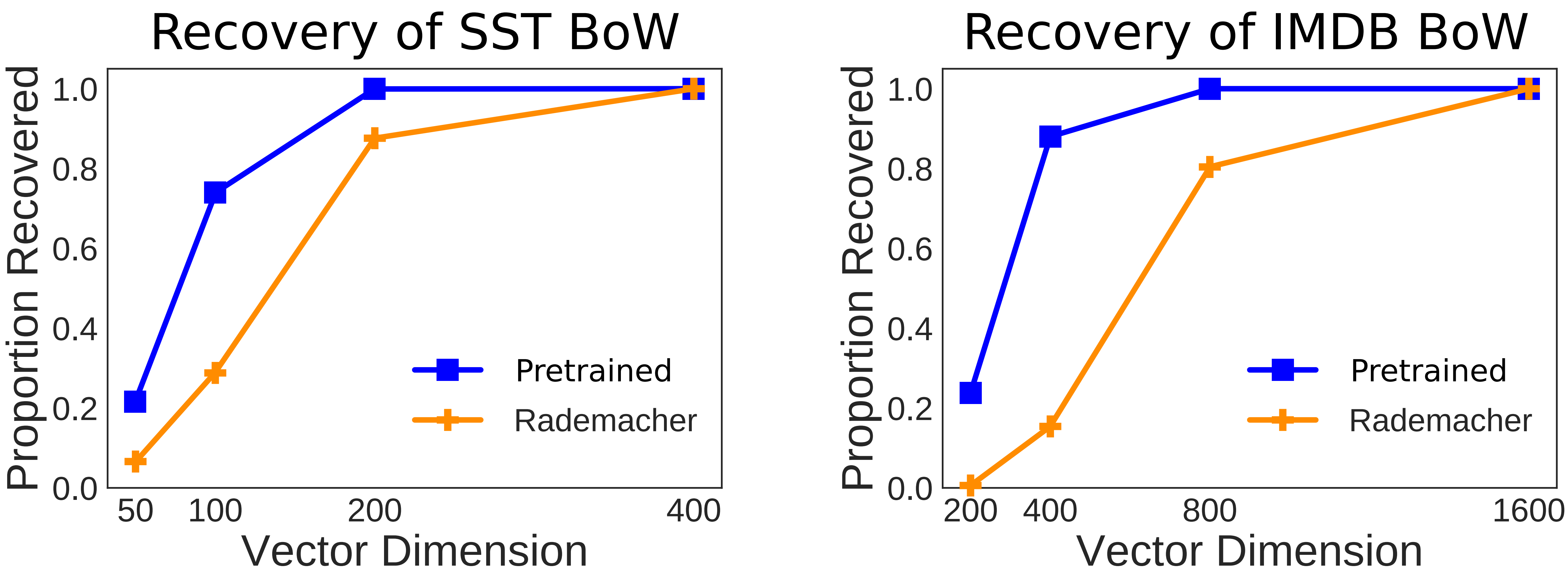
Can compressed sensing theory explain word embedding recovery?

- RIP/incoherence approach is too strong
  - must hold for all sparse signals
  - requires vectors with low inner product
- Weaker conditions often hard to check
- **Supporting Hyperplane Property (SHP):**[5] if there is a hyperplane **h** containing the vectors of all words in a document and all other word vectors are in the same half-space as the origin then $\mathbf{x}_{\mathbf{BoW}}$ can be recovered from $\mathbf{A}\mathbf{x}_{\mathbf{BoW}}$ using l_1-minimization



5: Extends results by Donoho & Tanner (*PNAS* 2005).

Pretrained embeddings are more likely to satisfy SHP



Intuitive explanation: embedding objectives push words in the same document closer together through unsupervised learning over a large text corpus.

Word embeddings have nice properties; what about n-gram embeddings?

- Difficult to capture n-gram semantics with composition alone, especially element-wise multiplication.

- New method — *à la carte* embedding (ACL 2018):
  - Induces n-gram embeddings using corpus contexts
  - Computes the expected n-gram vector under a standard model for GloVe-like word embeddings
  - Even stronger performance on standard classification tasks:

Method	**MR**	**CR**	**SUBJ**	**MPQA**	**TREC**	**SST**	**SST**	**IMDB**
BonG	77.8	78.3	91.8	85.8	90.0	80.9	42.3	**89.8**
Sent2Vec[1]	76.3	79.1	91.2	87.2	85.8	80.2	31.0	85.5
skip-thought[2]	80.3	**83.8**	**94.2**	**88.9**	<u>**93.0**</u>	85.1	45.8	
SDAE[3]	74.6	78.0	90.8	86.9	78.4			
CNN-LSTM[4]	77.8	82.0	93.6	**89.4**	**92.6**			
MC-QT[5]	<u>**82.4**</u>	<u>**86.0**</u>	<u>**94.8**</u>	<u>**90.2**</u>	**92.4**	<u>**87.6**</u>		
à la carte	**81.8**	**84.3**	**93.8**	87.6	89.0	**86.7**	<u>**48.1**</u>	<u>**90.9**</u>

1: Pagliardini et al. ‘18, 2: Kiros et al. ‘15, 3: Hill et al. ’16, 4: Gan et al. ‘17, 5: Logeswaran and Lee ‘18

Discussion and Future Work

In theory — more mysteries of word embeddings:

- Good sparse recovery does not give provable guarantees for classification. Does compressed learning hold for conditions weaker than RIP?
- Is there a rigorous explanation for these properties for some objective/model?

In practice — simple methods are competitive with deep learning for unsupervised NLP:

- Are standard tasks too simple and/or noisy?
- Simplified approaches can lead to similar insights for other neural systems, both in NLP and beyond.

Thank you!

Paper available on OpenReview (ICLR 2018):

https://openreview.net/pdf?id=B1e5ef-C-

Contact:

{arora,mkhodak,nsaunshi}@cs.princeton.edu

kiran.vodrahalli@columbia.edu

Questions?