On the Connections between Compressed Sensing, Learning and Natural Language Representations

Nikunj Umesh Saunshi

A Master's Thesis
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Master of Science in Engineering

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Professor Sanjeev Arora

June 2018

Abstract

Low-dimensional vector embeddings, computed using LSTMs or simpler techniques, are a popular approach for capturing the "meaning" of text and a form of unsupervised learning useful for downstream tasks. However, their power is not theoretically understood. The current paper derives formal understanding by looking at the subcase of linear embedding schemes. Using the theory of compressed sensing we show that representations combining the constituent word vectors are essentially information-preserving linear measurements of Bag-of-n-Grams (BonG) representations of text. This leads to a new theoretical result about LSTMs: low-dimensional embeddings derived from a low-memory LSTMs are provably at least as powerful on classification tasks, up to small error, as a linear classifier over BonG vectors, a result that extensive empirical work has thus far been unable to show. We also provide experimental evidence for the theoretical results by using random vectors for words. Furthermore using pretrained word embeddings such as GloVe and word2vec, we obtain strong, simple and unsupervised baselines on standard benchmarks and in some cases obtain state of the art performance among word-level methods.

Acknowledgements

Firstly I would like to thank my adviser, Prof. Sanjeev Arora, for his guidance over the last two years. I would also like to thank my collaborators, Misha Khodak and Kiran Vodrahalli for insightful discussions and making this thesis happen. This thesis was in part supported by NSF grants CCF-1302518 and CCF-1527371, Simons Investigator Award, Simons Collaboration Grant, and ONR-N00014-16-1-2329.

I would also like to thank professors Elad Hazan, Yoram Singer, Peter Ramadge, Samory Kpotufe, Emmanuel Abbe and all the teachers who taught me valuable skills through various courses and discussions. I am grateful to the Department for providing me the resources and the opportunity to be in the company of brilliant faculty and peers. Special thanks go to the Theory Lab visitors and ML lounge residents for all the exciting and intellectually stimulating discussions.

Finally, I would like to thank my parents and my sister for always being there for me and supporting me throughout my life, especially over the last two years.

To my parents and my sister

Contents

Abstract iii

Acknowledgements iv

List of Tables viii

List of Figures ix

1 Introduction **1**

2 Related Work **4**

3 Document Embeddings **6**

3.1 The Bag-of-n-Grams Vectors 7

3.2 Low-Dimensional n-Gram Embeddings 7

3.3 LSTM Representations 9

4 LSTMs as Compressed Learners **11**

4.1 Compressed Sensing and Learning 12

4.2 Proof of Main Result 14

5 Empirical findings **15**

5.1 Convergence to BonG 15

5.2 Performance on Tasks 16

5.3 Discussions 17

6 Conclusion **19**

6.1 Future Work 19

A Proof of Proposition 3.3.1 **20**

B Proof of Theorem 4.1.1 **22**

C Proof of Lemma 4.2.1 **27**

Bibliography **30**

List of Tables

5.1 Evaluation of DisC and recent unsupervised word-level approaches on standard classification tasks, with the character LSTM of [29] shown for comparison. The top three results for each dataset are **bolded**, the best is *italicized*, and the best word-level performance is underlined. We use normalized 1600-dimensional GloVe embeddings [27] trained on the Amazon Product Corpus [21] 17

5.2 Performance of DisC and other recent approaches on pairwise similarity and classification tasks. The top three results for each task are **bolded** and the best is underlined. 18

List of Figures

1.1 The pipeline for linear classification of unsupervised text representations. The transducer could be LSTMs, BonGs or simple linear schemes 3

5.1 IMDB performance of unigram (left) and bigram (right) DisC embeddings compared to the original dimension. 16

5.2 IMDB performance compared to training sample size. 18

5.3 Time needed to initialize model, construct document representations, and train a linear classifier on a 16-core compute node. 18

Chapter 1

Introduction

Much attention has been paid to using LSTMs [13] and similar recurrent models to compute text embeddings [4, 7]. Once trained, the LSTM can sweep once or twice through a given piece of text, process it using only limited memory and output a vector with moderate dimensionality (a few hundred to a few thousand), which can be used to measure text similarity via cosine similarity or as a featurization for downstream tasks.

The powers and limitations of this method have not been formally established. For example, can such neural embeddings compete with and replace traditional linear classifiers trained on trivial Bag-of-n-Grams (BonG) representations? Tweaked versions of BonG classifiers are known to be a surprisingly powerful baseline [33] and have fast implementations [15]. They continue to give better performance on many downstream supervised tasks such as IMDB sentiment classification [19] than purely unsupervised LSTM representations [18, 11, 23]. Even a very successful character-level (and thus computation-intensive, taking a month of training) approach does not reach BonG performance on datasets larger than IMDB [29]. Meanwhile there is evidence suggesting that simpler *linear* schemes give compact representations that provide most of the benefits of word-level LSTM embeddings [35, 3]. These linear

schemes consist of simply adding up, with a few modifications, standard pretrained word embeddings such as GloVe or word2vec [22, 27].

The current paper ties these disparate threads together by giving an information-theoretic account of linear text embeddings. We describe linear schemes that preserve n-gram information as low-dimensional embeddings with *provable* guarantees for their performance compared to that of the sparse representation on linear text classification task. The previous linear schemes, which used unigram information, are subcases of our approach, but our best schemes can also capture n-gram information with low additional overhead. Furthermore, the properties of word vectors used to prove good performance on classification tasks also imply recovery of the unigram information from the low-dimensional embedding. This suggests a deeper connection between classification performance and sparse recovery/compressed sensing [6]. Our approach also fits in the tradition of the older work on *distributed representations* of structured objects, especially the works of [28] and [16]. The following are the main results achieved by this new world-view:

1. Using random vectors as word embeddings in our linear scheme (instead of pretrained vectors) already allows us to rigorously show that low-memory LSTMs are *provably* at least as good as the full BonG vector on every linear classification task. This is a novel theoretical result in deep learning, obtained relatively easily using ideas from compressed sensing. By contrast, extensive empirical study of this issue has been inconclusive (apart from character-level models, and even then only on smaller datasets [29]). Note also that empirical work by its nature can only establish performance on some available datasets, not on *all* possible classification tasks. We prove this theorem in Section 4.2 by providing a nontrivial generalization of a result combining compressed sensing and learning [5]. In fact, before our work we do not know of any provable quantification of the power of any text embedding.

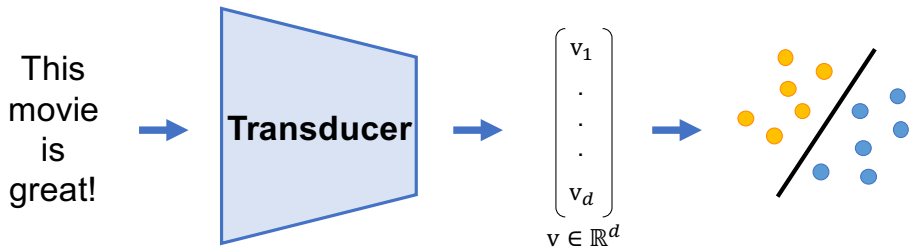

Figure 1.1: The pipeline for linear classification of unsupervised text representations. The transducer could be LSTMs, BonGs or simple linear schemes

2. We study experimentally how our linear embedding scheme improves when it uses pretrained embeddings (GloVe etc.) instead of random vectors. In addition we study empirically the effect of increasing the word embedding dimension on classification performance. Empirical results match results predicted by theory.

3. Section 5 provides empirical results supporting the above theoretical work, reporting accuracy of our linear schemes on multiple standard classification tasks. Our embeddings are consistently competitive with recent results and perform much better than all previous linear methods. Among unsupervised word-level representations they achieve state of the art performance on both the binary and fine-grained SST sentiment classification tasks [30]. Since our document representations are fast, compositional, and simple to implement given standard word embeddings, they provide strong baselines for future work.

This thesis is based on joint work with Sanjeev Arora, Mikhail Khodak and Kiran Vodrahalli [1] published at ICLR 2018.

Chapter 2

Related Work

Neural text embeddings are instances of *distributed representations*, long studied in connectionist approaches because they decay gracefully with noise and allow distributed processing. An early problem formulation for distributed representations was provided in [12] and [28] provided an elementary solution, the *holographic distributed representation*, which represents structured objects using circular vector convolution and has an easy and more compact implementation using the fast Fourier transform (FFT). Plate suggested applying such ideas to text, where "structure" can be quantified using parse trees and other graph structures. Our method is also closely related in form and composition to the sparse distributed memory system of [16]. The motivation for these methods was to recover components of the original text using these representations with simple operations. In the unigram case our embedding reduces to the familiar sum of word embeddings, which is known to be surprisingly powerful [35], and with a few tweaks even more so [3].

Representations of BonG vectors have been studied through the lens of compression by [26], who computed representations based on classical lossless compression algorithms using a linear program (LP) to reduce the number of features used. Though they work well on a few classification tasks, their embeddings are still high-

dimensional ($d > 100K$) and quite complicated to implement. In contrast, linear projection schemes are simpler, more compact, and can leverage readily available word embeddings. [23] also used a linear scheme, representing documents as an average of learned word and bigram embeddings. However, the motivation and benefits of encoding BonGs in low-dimensions are not made explicit. The novelty in the current paper is the connection of distributed representations to compressed sensing, which is concerned with recovering high-dimensional sparse signals $x \in \mathbb{R}^N$ from low-dimensional linear measurements Ax. We specifically study conditions on matrix $A \in \mathbb{R}^{d \times N}$ when this is possible and what they can say about performance on downstream tasks. We build upon the previous work of [5] to prove learning under compression in general settings.

Chapter 3

Document Embeddings

In this section we define the two types of representations that our analysis will relate:

1. high-dimensional sparse BonG vectors counting the occurrences of each k-gram for $k \leq n$

2. low-dimensional dense representations, from simple vector sums to novel n-gram-based embeddings and their concatenation

Although some of these representations have been previously studied and used, we define them so as to make clear their connection via compressed sensing, i.e. that representations of the second type are simply linear measurements of the first.

We now define some notation. Let V be the number of words in the vocabulary and V_n be the number of n-grams (independent of word order), so that $V = V_1$. Furthermore set $V_n^{\text{sum}} = \sum_{k \leq n} V_k$ and $V_n^{\max} = \max_{k \leq n} V_k$. We will use words/$n$-grams and indices interchangeably, e.g. if (a, b) is the ith of V_2 bigrams then the one-hot vector $e_{(a,b)}$ will be 1 at index i. Where necessary we will use $\{,\}$ to denote a multi-set and $(,)$ to denote a tuple. For any m vectors $v_i \in \mathbb{R}^d$ for $i = 1, \ldots, m$ we define $[v_1, \ldots, v_m]$ to be their concatenation, which is thus an element of $\mathbb{R}^{md}$. Finally, for any subset $\mathcal{X} \subset \mathbb{R}^N$ we denote by $\Delta\mathcal{X}$ the set $\{x - x' : x, x' \in \mathcal{X}\}$.

3.1 The Bag-of-n-Grams Vectors

Assigning to each word a unique index $i \in [V]$ we define the *Bag-of-Words* (BoW) representation x^{BoW} of a document to be the V-dimensional vector whose ith entry is the number of times word i occurs in the document. The n-gram extension of BoW is the *Bag-of-n-Grams* (BonG) representation, which counts the number of times any k-gram for $k \leq n$ appears in a document. Linear classification over such vectors has been found to be a strong baseline [33].

For ease of analysis we simplify the BonG approach by merging all n-grams in the vocabulary that contain the same words but in a different order. We call these features *n-cooccurrences* and find that the modification does not affect performance significantly. Formally for a document $w_1, \ldots, w_T$ we define the *Bag-of-n-Cooccurrences* (BonC) vector as the concatenation

$$x^{\text{BonC}} = \left[\sum_{t=1}^{T} e_{w_t} \quad , \quad \cdots \quad , \quad \sum_{t=1}^{T-n+1} e_{\{w_t, \ldots, w_{t+n-1}\}} \right] \tag{3.1}$$

which is thus a V_n^{sum}-dimensional vector. Note that for unigrams this is equivalent to the BoW vector.

3.2 Low-Dimensional n-Gram Embeddings

Now suppose each word w has a vector $v_w \in \mathbb{R}^d$ for some $d \ll V$. Then given a document $w_1, \ldots, w_T$ we define its *unigram embedding* as $z^{\text{u}} = \sum_{t=1}^{T} v_{w_t}$. While this is a simple and widely used featurization, we focus on the following straightforward relation with BoW: if $A \in \mathbb{R}^{d \times V}$ is a matrix whose columns are word vectors v_w then $Ax^{\text{BoW}} = \sum_{t=1}^{T} Ae_{w_t} = \sum_{t=1}^{T} v_{w_t} = z^{\text{u}}$. Thus in terms of compressed sensing the unigram embedding of a document is a d-dimensional linear measurement of its Bag-of-Words vector.

We could extend this unigram embedding to n-grams by first defining a representation for each n-gram as the tensor product of the vectors of its constituent words. Thus for each bigram $b = (w_1, w_2)$ we would have $v_b = v_{w_1} v_{w_2}^T$ and more generally $v_g = \bigotimes_{t=1}^{n} v_{w_t}$ for each n-gram $g = (w_1, \ldots, w_n)$. The document embedding would then be the sum of the tensor representations of all n-grams.

The major drawback of this approach is of course the blowup in dimension – n-grams are d^n dimensional – which in practice prevents its use beyond $n = 2$. To combat this a low-dimensional sketch or projection of the tensor product can be used, such as the circular convolution operator of [28]. Since we are interested in representations that can also be constructed by a low memory LSTM, we instead sketch this tensor product using the element-wise multiplication operation, which we find also usually works better than circular convolution in practice. Thus for the n-cooccurrence $g = \{w_1, \ldots, w_n\}$, we define the *distributed cooccurrence* (DisC) embedding $\tilde{v}_g = d^{\frac{n-1}{2}} \bigodot_{t=1}^{n} v_{w_t}$. The coefficient is required when the vectors v_w are random and unit norm to ensure that the product also has close to unit norm. In addition to their convenient form, DisC embeddings have nice theoretical and practical properties: they preserve the original embedding dimension, they reduce to unigram (word) embeddings for $n = 1$, and under mild assumptions they satisfy useful compressed sensing properties with overwhelming probability (Lemma 4.2.1).

We define the DisC document embedding to be the nd-dimensional weighted concatenation, over $k \leq n$, of the sum of the DisC vectors of all k-grams in a document:

$$z^{(n)} = \left[C_1 \sum_{t=1}^{T} \tilde{v}_{w_t} \quad , \quad \ldots \quad , \quad C_n \sum_{t=1}^{T-n+1} \tilde{v}_{\{w_t, \ldots, w_{t+n-1}\}} \right] \tag{3.2}$$

Here scaling factors C_k are set so that all spans of d coordinates have roughly equal norm (for random embeddings $C_k = 1$; for word embeddings $C_k = 1/k$ works well). Note that since $\tilde{v}_{w_t} = v_{w_t}$ we have $z^{(1)} = z^{\mathrm{u}}$ in the unigram case. Furthermore, as

with unigram embeddings by comparing (3.1) and (3.2) one can easily construct a $\sum_{k=1}^{n} dn \times V_n^{\text{sum}}$ matrix $A^{(n)}$ such that $z^{(n)} = A^{(n)} x^{\text{BonC}}$.

3.3 LSTM Representations

As discussed previously, LSTMs have become a common way to apply the expressive power of RNNs, with success on a variety of classification, representation, and sequence-to-sequence tasks. For document representation, starting with $h_0 = \mathbf{0}_m$ an *m-memory LSTM initialized with word vectors* $v_w \in \mathbb{R}^d$ takes in words $w_1, \ldots, w_T$ one-by-one and computes the document representation

$$h_t = f(\mathcal{T}_f(v_{w_t}, h_{t-1})) \circ h_{t-1} \quad + \quad i(\mathcal{T}_i(v_{w_t}, h_{t-1})) \circ g(\mathcal{T}_g(v_{w_t}, h_{t-1})) \tag{3.3}$$

where $h_t \in \mathbb{R}^m$ is the *hidden representation at time t*, the *forget gate* f, *input gate* i, and *input function* g are a.e. differentiable nondecreasing elementwise "activation" functions $\mathbb{R}^m \mapsto \mathbb{R}^m$, and affine transformations $\mathcal{T}_*(x, y) = W_* x + U_* y + b_*$ have weight matrices $W_* \in \mathbb{R}^{m \times d}, U_* \in \mathbb{R}^{m \times m}$ and bias vectors $b_* \in \mathbb{R}^m$. The *LSTM representation* of a document is then the state at the last time step, i.e. $z^{\text{LSTM}} = h_T$. Note that we will follow the convention of using *LSTM memory* to refer to the dimensionality of the hidden states. Since the LSTM is initialized with an embedding for each word it requires $\mathcal{O}(m^2 + md + Vd)$ computer memory, but the last term is just a lookup table so the vocabulary size does not factor into iteration or representation complexity.

From our description of LSTMs it is intuitive to see that one can initialize the gates and input functions so as to construct the DisC embeddings defined in the previous section. We state this formally and give the proof in the unigram case (the full proof appears in Appendix A):

Proposition 3.3.1. *Given word vectors $v_w \in \mathbb{R}^d$, one can initialize an $\mathcal{O}(nd)$-memory LSTM* (3.3) *that takes in words $w_1, \ldots, w_T$ (padded by an end-of-document token assigned vector $\mathbf{0}_d$) and constructs the DisC embedding* (3.2) *(up to zero padding), i.e. such that for all documents $z^{LSTM} = z^{(n)}$.*

Proof (Unigram Case). Set $f(x) = i(x) = g(x) = x$, $\mathcal{T}_f(v_{w_t}, h_{t-1}) = \mathcal{T}_i(v_{w_t}, h_{t-1}) = \mathbf{1}_d$, and $\mathcal{T}_g(v_{w_t}, h_{t-1}) = C_1 v_{w_t}$. Then $h_t = h_{t-1} + C_1 v_{w_t}$, so since $h_0 = \mathbf{0}_d$ we have the final LSTM representation $z^{\text{LSTM}} = h_T = C_1 \sum_{t=1}^{t} v_{w_t} = z^{(1)}$. □

By Proposition 3.3.1 we can construct a fixed LSTM that can compute compressed BonC representations on the fly and be further trained by stochastic gradient descent using the same memory.

Chapter 4

LSTMs as Compressed Learners

Our main contribution is to provide the first rigorous analysis of the performance of the text embeddings that we are aware of, showing that the embeddings of Section 3.2 can provide performance on downstream classification tasks at least as well any linear classifier over BonCs.

Theorem 4.0.1. *Let $S = \{(x_i, y_i)\}_{i=1}^m$ be drawn i.i.d. from a distribution $\mathcal{D}$ over BonC vectors of documents of length at most T satisfying assumptions 1 and 2 above and let w_0 be the linear classifier minimizing the logistic loss $\ell_{\mathcal{D}}$. Then for dimension $d = \Omega\left(\frac{T^2}{\varepsilon^2}\log\frac{nV_n^{max}}{\delta}\right)$ and appropriate choice of regularization coefficient one can initialize an $\mathcal{O}(nd)$-memory LSTM over i.i.d. word embeddings $v_w \sim \mathcal{U}^d\{\pm 1/\sqrt{d}\}$ such that w.p. $(1-\gamma)(1-2\delta)$ the classifier $\hat{w}$ minimizing the ℓ_2-regularized logistic loss over its representations satisfies*

$$\ell_{\mathcal{D}}(\hat{w}) \leq \ell_{\mathcal{D}}(w_0) + \mathcal{O}\left(\|w_0\|_2 \sqrt{\varepsilon + \frac{1}{m}\log\frac{1}{\delta}}\right) \tag{4.1}$$

We make two mild simplifying assumptions on the BonC vectors for the theorem:

1. The vectors are scaled by $\frac{1}{T\sqrt{n}}$, where T is the maximum document length. This assumption is made without loss of generality.

2. No n-cooccurrence contains a word more than once. While this is (infrequently) violated in practice, the problem can be circumvented by merging words as a preprocessing step.

The above theoretical bound shows that LSTMs match BonC performance as $\varepsilon \to 0$, which can be realized by increasing the embedding dimension d (c.f. Figure 5.1).

4.1 Compressed Sensing and Learning

Compressed sensing is concerned with recovering a high-dimensional k-sparse signal $x \in \mathbb{R}^N$ from a few linear measurements; given a design matrix $A \in \mathbb{R}^{d \times N}$ this is formulated as

$$\text{minimize} \quad \|w\|_0 \quad \text{subject to} \quad Aw = z \tag{4.2}$$

where $z = Ax$ is the *measurement vector*. As l_0-minimization is NP-hard, research has focused on sufficient conditions for tractable recovery. One such condition is the *Restricted Isometry Property* (RIP), for which [6] proved that (4.2) can be solved by convex relaxation:

Definition 4.1.1. *$A \in \mathbb{R}^{d \times N}$ is $(\mathcal{X}, \varepsilon)$-RIP for some subset $\mathcal{X} \subset \mathbb{R}^N$ if $\forall\ x \in \mathcal{X}$*

$$(1 - \varepsilon)\|x\|_2 \leq \|Ax\|_2 \leq (1 + \varepsilon)\|x\|_2 \tag{4.3}$$

We will abuse notation and say (k, ε)-RIP when $\mathcal{X}$ is the set of k-sparse vectors. This is the more common definition, but ours allows a more general Theorem 4.1.1 and a tighter bound in Theorem 4.0.1.

Following these breakthroughs, [5] studied whether it is possible to use the low-dimensional output of compressed sensing as a surrogate representation for classification. They proved a learning-theoretic bound on the loss of an SVM classifier in the

compressed domain compared to the best classifier in the original domain. In this work we are interested in comparing the performance of LSTMs with BonC representations, so we need to generalize the [5] result to handle Lipschitz losses and an arbitrary set $\mathcal{X} \subset \mathbb{R}^N$ of high-dimensional signals:

Theorem 4.1.1. *For any subset $\mathcal{X} \subset \mathbb{R}^N$ containing the origin let $A \in \mathbb{R}^{d \times N}$ be $(\Delta\mathcal{X}, \varepsilon)$-RIP and let m samples $S = \{(x_i, y_i)\}_{i=1}^m \subset \mathcal{X} \times \{-1, 1\}$ be drawn i.i.d. from some distribution $\mathcal{D}$ over $\mathcal{X}$ with $\|x\|_2 \leq R$. If ℓ is a λ-Lipschitz convex loss function and $w_0 \in \mathbb{R}^N$ is its minimizer over $\mathcal{D}$ then w.p. $1 - 2\delta$ the linear classifier $\hat{w}_A \in \mathbb{R}^d$ minimizing the ℓ_2-regularized empirical loss function $\ell_{S_A}(w) + \frac{1}{2C}\|w\|_2^2$ over the compressed sample $S_A = \{(Ax_i, y_i)\}_{i=1}^m \subset \mathbb{R}^d \times \{-1, 1\}$ satisfies*

$$\ell_{\mathcal{D}}(\hat{w}_A) \leq \ell_{\mathcal{D}}(w_0) + \mathcal{O}\left(\lambda R \|w_0\|_2 \sqrt{\varepsilon + \frac{1}{m}\log\frac{1}{\delta}}\right) \tag{4.4}$$

for appropriate choice of C. Recall that $\Delta\mathcal{X} = \{x - x' : x, x' \in \mathcal{X}\}$ for any $\mathcal{X} \subset \mathbb{R}^N$.

While a detailed proof of this theorem is spelled out in Appendix B, the main idea is to compare the distributional loss incurred by a classifier $\hat{w}$ in the original space to the loss incurred by $A\hat{w}$ in the compressed space. We show that the minimizer of the regularized empirical loss in the original space ($\hat{w}$) is a bounded-coefficient linear combination of samples in S, so its loss depends only on inner products between points in $\mathcal{X}$. Thus using RIP and a generalization error result by [31] we can bound the loss of $\hat{w}_A$, the regularized classifier in the compressed domain. Note that to get back from Theorem 4.1.1 the $\mathcal{O}(\sqrt{\varepsilon})$ bound for k-sparse inputs of [5] we can set $\mathcal{X}$ to the be the set of k-sparse vectors and assume A is $(2k, \varepsilon)$-RIP.

4.2 Proof of Main Result

To apply Theorem 4.1.1 we need the design matrix $A^{(n)}$ transforming BonCs into the DisC embeddings of Section 3.2 to satisfy the following RIP condition (Lemma 4.2.1), which we prove using a restricted isometry result for structured random sampling matrices in Appendix C:

Lemma 4.2.1. *Assume the setting of Theorem 4.0.1 and let $A^{(n)}$ be the $nd \times V_n^{sum}$ matrix relating DisC and BonC representations of any document by $z^{(n)} = A^{(n)} x^{BonC}$. If $d = \Omega\left(\frac{T^2}{\varepsilon^2} \log \frac{nV_n^{max}}{\delta}\right)$ then $A^{(n)}$ is $\left(\Delta\mathcal{X}_T^{(n)}, \varepsilon\right)$-RIP w.p. $1-\gamma$, where $\mathcal{X}_T^{(n)}$ is the set of BonCs of documents of length at most T.*

Proof of Theorem 4.0.1. Let $\hat{S} = \{(A^{(n)}x_i, y_i) : (x_i, y_i) \in S\}$, where $A^{(n)}$ is as in Lemma 4.2.1. Then by the same lemma $A^{(n)}$ is $\left(\Delta\mathcal{X}_T^{(n)}, \varepsilon\right)$-RIP w.p. $1-\gamma$, where $\mathcal{X}_T^{(n)}$ is the set of BonC vectors of documents of length at most T. By BonC assumption (1) all BonCs lie within the unit ball, so we can apply Theorem 4.1.1 with ℓ the logistic loss, $\lambda = 1$, and $R = 1$ to get that a classifier $\hat{w}$ trained using ℓ_2-regularized logistic loss over $\hat{S}$ will satisfy the required bound (4.1). Since by Proposition 3.3.1 one can initialize an $\mathcal{O}(nd)$-memory LSTM that takes in i.i.d. Rademacher word vectors $v_w \sim \mathcal{U}^d\{\pm 1/\sqrt{d}\}$ such that $z^{\text{LSTM}} = z^{(n)} = A^{(n)}x \;\forall\; x \in \mathcal{X}_T^{(n)}$, this completes the proof. □

Chapter 5

Empirical findings

Our theoretical results show that simple tensor product sketch-based n-gram embeddings can approach BonG performance and be computed by a low-memory LSTM. In this section we compare these text representations and others on several standard tasks, verifying that DisC performance approaches that of BonCs as dimensionality increases and establishing several baselines for text classification. Code to reproduce results is provided at `https://github.com/NLPrinceton/text_embedding`.

5.1 Convergence to BonG

We first analyze empirically how well our model approximates BonC performance. As predicted by Theorem 4.0.1, the performance of random embeddings on IMDB movie reviews classification [19] approaches that of BonC as dimension increases and the isometry distortion ε decreases (Figure 5.1). In practice however, semantic word embeddings such as GloVe [27], word2vec [22], SN [2] that preserve the "meaning" of words have been successful at various NLP tasks such as analogies and improve the performance of language models and various classification tasks. Even though Theorem 4.0.1 says nothing about these pretrained embeddings, DisC embeddings

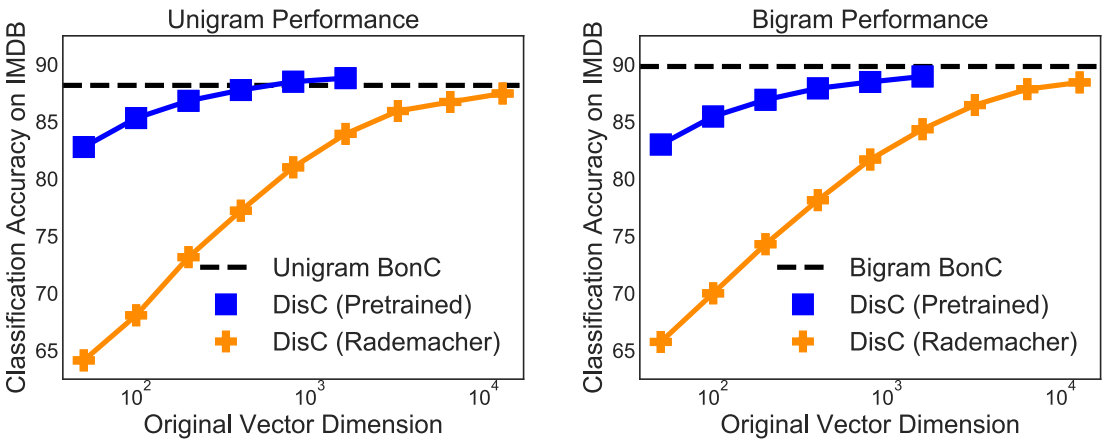

Figure 5.1: IMDB performance of unigram (left) and bigram (right) DisC embeddings compared to the original dimension.

constructed using these SN embeddings approach BonC performance much earlier, surpassing it in the unigram case.

5.2 Performance on Tasks

We test classification on MR movie reviews [25], CR customer reviews [14], SUBJ subjectivity dataset [24], MPQA opinion polarity subtask [34], SST sentiment classification (binary and fine-grained) [30], and IMDB movie reviews [19]. The first four are evaluated using 10-fold cross-validation, while the others have train-test splits. In all cases we use logistic regression with ℓ_2-regularization determined by cross-validation. We further test DisC on the SICK relatedness and entailment tasks [20] and the MRPC paraphrase detection task [8]. The inputs here are sentences pairs (a, b) and the standard featurization for document embeddings x_a and x_b of a and b is $[|x_a - x_b|, x_a \odot x_b]$ [32]. We use logistic regression for SICK entailment and MRPC and use ridge regression to predict similarity scores for SICK relatedness, with ℓ_2-regularization determined by cross-validation. Since BonGs are not used for pairwise tasks our theory says nothing about performance here; we include these evaluations to demonstrate the versatility of our representations.

Representation	n	d^*	MR	CR	SUBJ	MPQA	SST	SST†	IMDB
BonC (3.1)	1	V_1	77.1	77.0	91.0	85.1	80.7	36.8	88.3
	2	V_2^{sum}	77.8	78.1	91.8	85.8	80.9	39.0	**90.0**
	3	V_3^{sum}	77.8	78.3	91.4	85.6	80.1	42.3	89.8
DisC (3.2)	1	1600	79.6	81.0	92.4	87.8	84.6	45.7	89.2
	2	3200	**80.1**	81.5	92.6	87.9	**85.5**	**46.4**	89.4
	3	4800	80.0	81.3	92.6	87.9	**85.2**	**46.7**	89.6
SIF[1]	1	1600	79.6	81.1	92.5	87.7	84.4	45.8	89.2
Sent2Vec[2]	1	700	76.2	78.7	91.2	87.2	80.2	31.0	85.5
Sent2Vec[2]	2	700	76.3	79.1	91.1	86.6	80.0	30.7	85.3
CFL[3]	5	100K+							**90.4**
Paragraph Vec.[4]			74.8	78.1	90.5	74.2			
skip-thoughts[4]		4800	**80.3**	**83.8**	**94.2**	**88.9**	85.1	45.8	
SDAE[5]		2400	74.6	78.0	90.8	86.9			
CNN-LSTM[6]		4800	77.8	**82.0**	**93.6**	***89.4***			
byte mLSTM[7]		4096	***86.8***	***90.6***	***94.7***	**88.8**	***91.7***	***54.6***	***92.2***

* Vocabulary sizes (i.e. BonC dimensions) vary by task; usually 10K-100K.

† Fined-grained task with 5 classes.

[1] [3] Reported performance of best hyperparameter using Amazon GloVe embeddings.

[2,4,7] [23, 18, 29] Evaluated latest pretrained models. Note that the available skip-thoughts implementation fails on the IMDB and MRPC tasks

[3,5,6] [26, 11, 10] From publication (+emb version of last two).

Table 5.1: Evaluation of DisC and recent unsupervised word-level approaches on standard classification tasks, with the character LSTM of [29] shown for comparison. The top three results for each dataset are **bolded**, the best is *italicized*, and the best word-level performance is underlined. We use normalized 1600-dimensional GloVe embeddings [27] trained on the Amazon Product Corpus [21]

5.3 Discussions

We find that DisC representation performs consistently well relative to recent unsupervised methods; among word-level approaches it is the top performer on the SST tasks and competes on many others with skip-thoughts and CNN-LSTM, both concatenations of two LSTM representations. Our method does not require extravagant computing resources unlike LSTM representations. It is useful as a strong baseline, often beating BonCs and many more complicated approaches while taking much less time to represent and train on documents than neural representations (Figure 5.3).

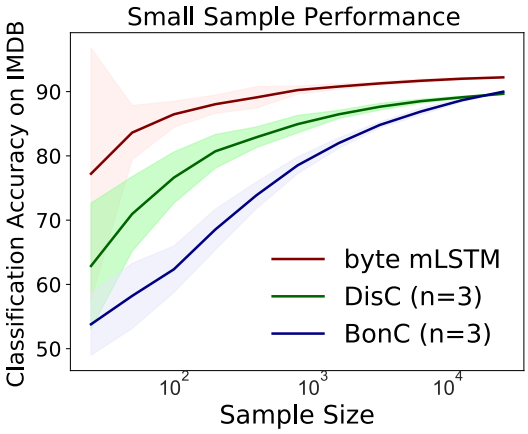

Figure 5.2: IMDB performance compared to training sample size.

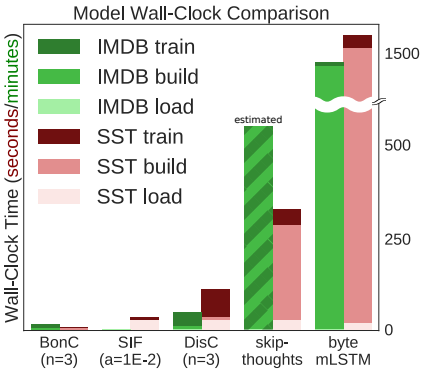

Figure 5.3: Time needed to initialize model, construct document representations, and train a linear classifier on a 16-core compute node.

Rep.	n	SICK-R (r/ρ)	SICK-E	MRPC (Acc./F_1)
DisC (3.2)	1	73.6 / 71.0	81.9	73.1 / **81.8**
	2	75.6 / 72.1	**83.2**	70.8 / 79.0
	3	**76.2 / 72.2**	**82.5**	73.1 / 81.6
SIF	1	73.7 / 68.5	82.4	73.5 / **82.1**
Sent2Vec	1	69.3 / 64.6	78.6	71.3 / 80.7
Sent2Vec	2	70.1 / 65.3	78.7	70.0 / 79.3
skip-thoughts		**82.4** / **76.0**	**83.2**	
SDAE				**73.7 / 80.7**
CNN-LSTM				**76.4** / **83.8**
byte mLSTM		**78.5 / 72.1**	80.0	**73.8** / 81.4

Model information is the same as that in Table 5.1.

Table 5.2: Performance of DisC and other recent approaches on pairwise similarity and classification tasks. The top three results for each task are **bolded** and the best is underlined.

Chapter 6

Conclusion

In this paper we explored the connection between compressed sensing, learning, and natural language representation. We first related LSTM and BonG methods via word embeddings, coming up with simple new document embeddings based on tensor product sketches. Then we studied their classification performance, proving a generalization of the compressed learning result of [5] to convex Lipschitz losses and a bound on the loss of a low-dimensional LSTM classifier in terms of its (modified) BonG counterpart, an issue which neither experiments nor theory had been able to resolve. Finally, we observed that using pretrained embeddings in DisC converge much faster than random embeddings to BonG performance.

6.1 Future Work

The n-gram embeddings we suggest are simple, compositional and can be analyzed formally. However these might not be the optimal embeddings for classification and exploring simple methods to construct better n-gram embeddings [23], [17] could be direction for future. Extending the theory to the case of pretrained word embeddings (which are used extensively in practice) could give us a better understanding of our current systems and could potentially guide us to better text representations.

Appendix A

Proof of Proposition 3.3.1

Let $f(x) = i(x) = g(x) = x$ with

$$\mathcal{T}_f(v_{w_t}, h_{t-1}) = \begin{pmatrix} \mathbf{1}_{nd} \\ \mathbf{0}_{(n-1)d} \end{pmatrix}$$

$$\mathcal{T}_i(v_{w_t}, h_{t-1}) = \begin{pmatrix} \mathbf{0}_{d\times nd} & \cdots & \mathbf{0}_{d\times d} \\ \vdots & I_{(n-2)d} & \mathbf{0}_{(n-2)d\times d} \\ \vdots & \ddots & I_d \\ \vdots & \ddots & \mathbf{0}_{d\times d} \\ \mathbf{0}_{(n-2)d\times nd} & I_{(n-2)d} & \mathbf{0}_{(n-2)d\times d} \end{pmatrix} h_{t-1} + \begin{pmatrix} \mathbf{1}_d \\ \mathbf{0}_{(n-1)d} \\ \mathbf{1}_d \\ \mathbf{0}_{(n-2)d} \end{pmatrix}$$

$$\mathcal{T}_g(v_{w_t}, h_{t-1}) = \begin{pmatrix} C_1 I_d \\ \vdots \\ C_n d^{\frac{n-1}{2}} I_d \\ I_d \\ \vdots \\ I_d \end{pmatrix} v_{w_t}$$

Substituting these parameters into the LSTM update (3.3) and using $h_0 = 0$ we have $\forall\ t > 0$ that

$$h_t = \begin{pmatrix} C_1 \sum_{\tau=1}^{t} v_{w_\tau} \\ \vdots \\ C_n d^{\frac{n-1}{2}} \sum_{\tau=1}^{t-n+1} \bigodot_{k=1}^{n} v_{w_{\tau+k-1}} \\ v_{w_t} \\ \vdots \\ \bigodot_{k=1}^{n-1} v_{w_{t+k-n+1}} \end{pmatrix} = \begin{pmatrix} C_1 \sum_{\tau=1}^{t} \tilde{v}_{w_\tau} \\ \vdots \\ C_n d^{\frac{n-1}{2}} \sum_{\tau=1}^{t-n+1} \tilde{v}_{\{w_\tau,\ldots,w_{\tau+n-1}\}} \\ \tilde{v}_{w_t} \\ \vdots \\ \tilde{v}_{\{w_{t-n+2},\ldots,w_t\}} \end{pmatrix}$$

Thus

$$h_T = \begin{pmatrix} C_1 \sum_{t=1}^{T} \tilde{v}_{w_t} \\ \vdots \\ C_n d^{\frac{n-1}{2}} \sum_{t=1}^{T-n+1} \tilde{v}_{\{w_t,\ldots,w_{t+n-1}\}} \\ \tilde{v}_{w_T} \\ \vdots \\ \tilde{v}_{\{w_{T-n+2},\ldots,w_T\}} \end{pmatrix} = \begin{pmatrix} \tilde{z}^{(n)} \\ \tilde{v}_{w_T} \\ \vdots \\ \tilde{v}_{\{w_{T-n+2},\ldots,w_T\}} \end{pmatrix}$$

Note that $h_t \in \mathbb{R}^{(2n-1)d}$ so as desired the LSTM has $\mathcal{O}(nd)$-memory. Although h_T contains $(n-1)d$ more dimensions than $\tilde{z}^{(n)}$, by padding the end of the document with an end-of-document token whose word vector is $\mathbf{0}_d$ the entries in those dimensions will be set to zero by the update at the last step. Thus up to zero padding we will have $z^{\text{LSTM}} = h_T = \tilde{z}^{(n)}$.

Appendix B

Proof of Theorem 4.1.1

Throughout this section we assume the setting described in Theorem 4.1.1. Furthermore for some constant $C > 0$ define the ℓ_2-regularization of the loss function ℓ as

$$L(w) = \ell(w) + \frac{1}{2C}\|w\|_2^2$$

Lemma B.0.1. *Let $\hat{w}$ be the classifier obtained minimizing $L_S(w) = \frac{1}{m}\sum_{i=1}^{m} \ell(w^T x_i, y_i) + \frac{1}{2C}\|w\|_2^2$, where $\ell(\cdot,\cdot)$ is a convex λ-Lipschitz function in the first cordinate. Then*

$$\hat{w} = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{B.1}$$

where $|\alpha_i| \leq \frac{\lambda C}{m}\ \forall\ i$. This result holds in the compressed domain as well.

Proof. If ℓ is an λ-Lipschitz function, its sub-gradient at every point is bounded by λ. So by convexity, the unique optimizer is given by taking first-order conditions:

$$0 = \partial_w L_S(w) = \frac{w}{C} + \frac{1}{m}\sum_{i=1}^{m} \partial_{w^T x_i}\ell(w^T x_i, y_i)x_i$$

$$\implies \hat{w} = \frac{C}{m}\sum_{i=1}^{m} -y_i \partial_{\hat{w}^T x_i}\ell(\hat{w}^T x_i, y_i)y_i x_i \tag{B.2}$$

Since ℓ is Lipschitz, $|\partial_{w^T x_i} \ell(w^T x_i, y_i)| \leq \lambda$. Therefore the first-order optimal solution (B.2) of $\hat{w}$ can be expressed as (B.1) for some $\alpha_1, \ldots, \alpha_m$ satisfying $|\alpha_i| \leq \frac{\lambda C}{m} \ \forall \ i$, which is the desired result.

□

Lemma B.0.2. $x, x' \in \mathcal{X} \implies (1+\varepsilon)x^T x' - 2R^2\varepsilon \leq (Ax)^T(Ax') \leq (1-\varepsilon)x^T x' + 2R^2\varepsilon$

Proof. Since A is $(\Delta\mathcal{X}, \varepsilon)$-RIP we have $(1-\varepsilon)\|x-x'\|_2 \leq \|A(x-x')\|_2 \leq (1+\varepsilon)\|x-x'\|_2$. Also since $\mathbf{0}_N \in \mathcal{X}$, A is also $(\mathcal{X}, \varepsilon)$-RIP and the result then follows by the same argument as in [5, Lemma 4.2-3]. □

Corollary B.0.1. $\|\hat{w}\|_2^2 \leq \lambda^2 C^2 R^2$ *and* $\|\hat{w}_A\|_2^2 \leq \lambda^2 C^2 (1+\varepsilon)^2 R^2$.

Proof. The first bound follows by expanding $\|\hat{w}\|_2^2$ and using $\|x\|_2 \leq R$; the second follows by expanding $\|\hat{w}_A\|_2^2$, applying Lemma B.0.2 to bound inner product distortion, and using $\|x\|_2 \leq R$. □

Lemma B.0.3. *Let $\hat{w}$ be the linear classifier minimizing L_S. Then*

$$L_{\mathcal{D}}(A\hat{w}) \leq L_{\mathcal{D}}(\hat{w}) + \mathcal{O}(\lambda^2 C R^2 \varepsilon)$$

Proof. By Lemma B.0.1 we can re-express $\hat{w}$ using Equation B.1 and then apply the inequality from Lemma B.0.2 to get

$$\begin{aligned}
(A\hat{w})^T(Ax) &= \sum_{i=1}^{m} \alpha_i y_i (Ax_i)^T (Ax) \\
&\leq \sum_{i:\alpha_i y_i \geq 0} \alpha_i y_i \left((1-\varepsilon)x_i^T x + 2R^2\varepsilon\right) + \sum_{i:\alpha_i y_i < 0} \alpha_i y_i \left((1+\varepsilon)x_i^T x - 2R^2\varepsilon\right) \\
&= \hat{w}^T x - \varepsilon \sum_{i=1}^{m} |\alpha_i y_i| x_i^T x + 2R^2\varepsilon \sum_{i=1}^{m} |\alpha_i y_i| \leq \hat{w}^T x + 3\lambda C R^2 \varepsilon
\end{aligned}$$

$$
\begin{aligned}
(A\hat{w})^T(Ax) &= \sum_{i=1}^{m} \alpha_i y_i (Ax_i)^T (Ax) \\
&\geq \sum_{i:\alpha_i y_i \geq 0} \alpha_i y_i \left((1+\varepsilon) x_i^T x - 2R^2\varepsilon\right) + \sum_{i:\alpha_i y_i < 0} \alpha_i y_i \left((1-\varepsilon) x_i^T x + 2R^2\varepsilon\right) \\
&= \hat{w}^T x + \varepsilon \sum_{i=1}^{m} |\alpha_i y_i| x_i^T x - 2R^2\varepsilon \sum_{i=1}^{m} |\alpha_i y_i| \geq \hat{w}^T x - 3\lambda C R^2 \varepsilon
\end{aligned}
$$

for any $x \in \mathbb{R}^N$. Since ℓ is λ-Lipschitz taking expectations over $\mathcal{D}$ implies

$$
\ell_{\mathcal{D}}(A\hat{w}) \leq \ell_{\mathcal{D}}(\hat{w}) + 3\lambda^2 C R^2 \varepsilon \tag{B.3}
$$

Substituting Equation B.1 applying Lemma B.0.2 also yields

$$
\begin{aligned}
\|A\hat{w}\|_2^2 &= \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (Ax_i)^T (Ax_j) \\
&\leq \sum_{i,j:\alpha_i \alpha_j y_i y_j \geq 0} \alpha_i \alpha_j y_i y_j \left((1-\varepsilon) x_i^T x_j + 2R^2\varepsilon\right) \\
&\quad + \sum_{i,j:\alpha_i \alpha_j y_i y_j < 0} \alpha_i \alpha_j y_i y_j \left((1+\varepsilon) x_i^T x_j - 2R^2\varepsilon\right) \\
&\leq \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i,j} -|\alpha_i \alpha_j y_i y_j| \varepsilon x_i^T x_j + 2R^2 |\alpha_i \alpha_j y_i y_j| \varepsilon \\
&\leq \|\hat{w}\|_2^2 + 3\lambda^2 C^2 R^2 \varepsilon
\end{aligned}
$$

which implies

$$
\frac{1}{2C}\|A\hat{w}\|_2^2 \leq \frac{1}{2C}\|\hat{w}\|_2^2 + \frac{3}{2}\lambda^2 C R^2 \varepsilon \tag{B.4}
$$

Together the inequalities bounding the loss term (B.3) and the regularization term (B.4) imply the result.

□

Lemma B.0.4. *Let $\hat{w}$ be the linear classifier minimizing L_S and let w^* be the linear classifier minimizing $L_{\mathcal{D}}$. Then with probability $1-\gamma$*

$$L_{\mathcal{D}}(\hat{w}) \leq L_{\mathcal{D}}(w^*) + \mathcal{O}\left(\frac{\lambda^2 C R^2}{m} \log \frac{1}{\gamma}\right)$$

This result holds in the compressed domain as well.

Proof. By Corollary B.0.1 we have that $\hat{w}$ is contained in a closed convex subset independent of S. Therefore since ℓ is λ-Lipschitz, L is $\frac{1}{C}$-strongly convex, and $\|x\|_2 \leq \mathcal{O}(R)$, we have by [31, Theorem 1] that with probability $1-\gamma$

$$L_{\mathcal{D}}(\hat{w}) - L_{\mathcal{D}}(w^*) \leq 2\left[L_S(\hat{w}) - L_S(w^*)\right]_+ + \mathcal{O}\left(\frac{\lambda^2 C R^2}{m} \log \frac{1}{\gamma}\right)$$

Then since by definition $\hat{w}$ minimizes $L_S(w)$ we have that $L_S(\hat{w}) \leq L_S(w^*)$, which substituted into the previous equation completes the proof. □

Proof of Theorem 4.1.1. Applying Lemma B.0.4 in the compressed domain yields

$$\ell_{\mathcal{D}}(\hat{w}_A) \leq \ell_{\mathcal{D}}(\hat{w}_A) + \frac{1}{2C}\|\hat{w}_A\|_2^2 = L_{\mathcal{D}}(\hat{w}_A) \leq L_{\mathcal{D}}(w_A^*) + \mathcal{O}\left(\frac{\lambda^2 C R^2}{m} \log \frac{1}{\gamma}\right)$$

where w_A^* minimizes $L_{\mathcal{D}}$. By definition of w_A^*, $L_{\mathcal{D}}(w_A^*) \leq L_{\mathcal{D}}(A\hat{w})$, so together with Lemma B.0.3 and the previous inequality we have

$$\ell_{\mathcal{D}}(\hat{w}_A) \leq L_{\mathcal{D}}(A\hat{w}) + \mathcal{O}\left(\frac{\lambda^2 C R^2}{m} \log \frac{1}{\gamma}\right) \leq L_{\mathcal{D}}(\hat{w}) + \mathcal{O}\left(\lambda^2 C R^2 \left(\varepsilon + \frac{1}{m} \log \frac{1}{\gamma}\right)\right)$$

We now apply Lemma B.0.4 in the sparse domain to get

$$\ell_{\mathcal{D}}(\hat{w}_A) \leq L_{\mathcal{D}}(w^*) + \mathcal{O}\left(\lambda^2 C R^2 \left(\varepsilon + \frac{1}{m} \log \frac{1}{\gamma}\right)\right)$$

where w^* minimizes $L_{\mathcal{D}}$. By definition of w^*, $L_{\mathcal{D}}(w^*) \leq L_{\mathcal{D}}(w_0) = \ell_{\mathcal{D}}(w_0) + \frac{1}{2C}\|w_0\|_2^2$, so by the previous inequality we have

$$\ell_{\mathcal{D}}(\hat{w}_A) \leq \ell_{\mathcal{D}}(w_0) + \frac{1}{2C}\|w_0\|_2^2 + \mathcal{O}\left(\lambda^2 C R^2 \left(\varepsilon + \frac{1}{m}\log\frac{1}{\gamma}\right)\right)$$

Substituting the C that minimizes the r.h.s. of this inequality completes the proof. $\square$

Appendix C

Proof of Lemma 4.2.1

We assume the setting described in Lemma 4.2.1, where we are concerned with the RIP condition of the matrix $A^{(n)}$ when multiplying vectors $x \in \mathcal{X}_T^{(n)}$, the set of BonC vectors for documents of length at most T. This matrix can be written as

$$A^{(n)} = \begin{pmatrix} A_1 & \mathbf{0}_{d\times V_2} & \cdots & \mathbf{0}_{d\times V_n} \\ \mathbf{0}_{d\times V_1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{d\times V_n} \\ \mathbf{0}_{d\times V_1} & \cdots & \mathbf{0}_{d\times V_{n-1}} & A_n \end{pmatrix}$$

where A_p is the $d \times V_p$ matrix whose columns are the DisC embeddings of all p-grams in the vocabulary (and thus $A^{(1)} = A_1 = A$, the matrix of the original word embeddings). Note that from (3.1) any $x \in \mathcal{X}_T^{(n)}$ can be written as $x = [x_1, \ldots, x_n]$, where x_p is a T-sparse vector whose entries correspond to p-grams. Thus we also have $A^{(n)}x = [A_1x_1, \ldots, A_nx_n]$.

Lemma C.0.1. *If A_p is $(2k, \varepsilon)$-RIP w.p. $1-\gamma$ $\forall$ $p \in [n]$ then $A^{(n)}$ is $\left(\Delta\mathcal{X}_k^{(n)}, \varepsilon\right)$-RIP w.p. at least $1 - n\gamma$.*

Proof. By union bound we have that A_p is $(2k, \varepsilon)$-RIP $\forall\ p \in [n]$ with probability at least $1 - n\gamma$. Thus by Definition 4.1.1 we have w.p. $1 - n\gamma$ that $\forall\ x \in \Delta\mathcal{X}_k^{(n)}$

$$\|A^{(n)}x\|_2^2 = \sum_{p=1}^{n} \|A_p x_p\|_2^2 \leq \sum_{p=1}^{n} (1+\varepsilon)^2 \|x_p\|_2^2 = (1+\varepsilon)^2 \|x\|_2^2$$

Similarly, $\|A^{(n)}x\|_2^2 \geq (1-\varepsilon)^2\|x\|_2^2$. From Definition 4.1.1, taking the square root of both sides of both inequalities completes the proof. □

Definition C.0.1 ([9])**.** *Let $\mathcal{D}$ be a distribution over a subset $S \subset \mathbb{R}^n$. Then the set $\Phi = \{\phi_1, \ldots, \phi_N\}$ of functions $\phi_i : S \mapsto \mathbb{R}$ is a bounded orthonormal system (BOS) with constant B if we have $\mathbb{E}_{\mathcal{D}}(\phi_i\phi_j) = 1_{i=j}\ \forall\ i, j$ and $\sup_{s\in S} |\phi_i(s)| \leq B\ \forall\ i$. Note that by definition $B \geq 1$.*

Theorem C.0.1 ([9])**.** *If $d = \tilde{\Omega}\left(\frac{B^2 k}{\varepsilon^2} \log \frac{N}{\gamma}\right)$ for $(\varepsilon, \gamma) \in (0,1)$ and $\sqrt{d}A$ is a $d \times N$ matrix associated with a BOS with constant B then A is (k, ε)-RIP w.p. $1 - \gamma$.*

Lemma C.0.2. *If $d = \tilde{\Omega}\left(\frac{T}{\varepsilon^2} \log \frac{V_p}{\gamma}\right)$ and the word embeddings are drawn i.i.d. from $\mathcal{U}^d\{\pm 1/\sqrt{d}\}$ then for any $p \in [n]$ the matrix $A_p \in \mathbb{R}^{d \times V_p}$ of DisC embeddings is (T, ε)-RIP w.p. $1 - \gamma$.*

Proof. Note that by Theorem C.0.1 it suffices to show that $\sqrt{d}A_p$ is a random sampling matrix associated with a BOS with constant $B = 1$. Let $\mathcal{D} = \mathcal{U}^V\{\pm 1\}$ be the uniform distribution over V i.i.d. Rademacher random variables indexed by words in the vocabulary. Then by definition the matrix $A_p \in \mathbb{R}^{d \times V_p}$ can be constructed by drawing random variables $x^{(1)}, \ldots, x^{(d)}$ i.i.d. from $\mathcal{D}$ and assigning to the ijth entry of $\sqrt{d}A_p$ corresponding to the p-gram $g = \{g_1, \ldots, g_p\}$ the value $\phi_j\left(x^{(i)}\right) = \prod_{t=1}^{p} x_{g_t}^{(i)}$, where each function $\phi_j : \{\pm 1\}^V \mapsto \mathbb{R}$ is uniquely associated to its p-gram. It remains to be shown that this set of functions is a BOS with constant $B = 1$.

For any two p-grams g, g' and their functions ϕ_i, ϕ_j we have $\mathbb{E}_{\mathcal{D}}(\phi_i\phi_j) = \mathbb{E}_{x\sim\mathcal{D}}\left(\prod_{t=1}^{p} x_{g_t} x_{g'_t}\right)$, which will be 1 iff each word in $g \cup g'$ occurs an even number

of times in the product and 0 otherwise. Because all p-grams are uniquely defined under any permutation of its words (i.e. we are in fact using p-cooccurrences) and we have assumed that no p-gram contains a word more than once, each word occurs an even number of times in the product iff $g = g' \iff i = j$. Furthermore we have that $|\phi_i(x)| \leq 1 \; \forall \; x \in \{\pm 1\}^V \; \forall \; i$ by construction. Thus according to Definition C.0.1 the set of functions $\{\phi_1, \ldots, \phi_{V_p}\}$ associated to the p-grams in the vocabulary is a BOS with constant $B = 1$. □

Proof of Lemma 4.2.1. Since $d = \tilde{\Omega}\left(\frac{T}{\varepsilon^2} \log \frac{nV_n^{\max}}{\gamma}\right)$, Lemma C.0.2 implies that A_p is $(2T, \varepsilon)$-RIP w.p. $1 - \frac{\gamma}{n} \; \forall \; p \in [n]$. Applying Lemma C.0.1 yields the result. □

Bibliography

[1] Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and lstms. In *Proceedings of the International Conference on Learning Representations*, 2018.

[2] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the ACL*, 4:385–399, 2016.

[3] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the International Conference on Learning Representations*, 2017.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[5] Robert Calderbank, Sina Jafarpour, and Robert Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Technical report, 2009.

[6] Emmanuel Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51:4203–4215, 2005.

[7] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[8] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing*, 2005.

[9] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. 2013.

[10] Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin Carin. Learning generic sentence representations using convolutional neural networks. In *Proceedings of Empirical Methods in Natural Language Processing*, 2017.

[11] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the North American Chapter of the ACL*, 2016.

[12] Geoffrey Hinton. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46:47–75, 1990.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[14] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[15] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the ACL*, 2017.

[16] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1:139–159, 2009.

[17] Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, July 2018.

[18] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *Neural Information Processing Systems*, 2015.

[19] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*, 2011.

[20] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic model on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, 2014.

[21] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*, 2013.

[23] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. arXiv, 2017.

[24] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the ACL*, 2004.

[25] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the ACL*, 2005.

[26] Hristo S. Paskov, Robert West, John C. Mitchell, and Trevor J. Hastie. Compressive feature learning. In *Neural Information Processing Systems*, 2013.

[27] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

[28] Tony Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 1995.

[29] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment, 2017. arXiv.

[30] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of Empirical Methods in Natural Language Processing*, 2013.

[31] Karthik Sridharan, Nathan Srebro, and Shai Shalev-Schwartz. Fast rates for regularized objectives. In *Neural Information Processing Systems*. 2008.

[32] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the ACL*, 2015.

[33] Sida Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the ACL*, 2012.

[34] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39:165–210, 2005.

[35] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. In *Proceedings of the International Conference on Learning Representations*, 2016.