On the Compressed Sensing Properties of Word Embeddings

Mikhail Khodak

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

Distributed representations of words, or word embeddings, computed using large text corpora have become a popular way of encoding linguistic features for applications in natural language processing. However, their power, in terms of the information they encode and how this relates to performance on downstream tasks, is not theoretically understood. Drawing inspiration from results in compressed learning [7, 1], we present a remarkable empirical property of word embeddings - they are more efficient than random matrices for sparse recovery of Bag-of-Words vectors from linear compression. We discuss how this result can be understood by introducing a new, efficiently-verifiable compressed sensing property guaranteeing exact recovery of nonnegative signals that depends on geometric results connecting basis pursuit and neighborly polytopes [10]. Finally, we analyze the extent to which different embeddings satisfy this property and how to connect these results to understand the performance of these representations on downstream tasks.

Acknowledgements

I am very grateful for the patient guidance of my adviser, Professor Sanjeev Arora, over the past three years. I would also like to thank my collaborators, Nikunj Saunshi and Kiran Vodrahalli, and my second reader, Professor Yoram Singer, for their helpful insights. Finally, I am most grateful to my parents, sister, and grandmother for their support.

The work in 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.

Contents

Chapter 1

Introduction

Much attention has been paid to using LSTMs [14] and similar models to compute text embeddings [5, 9] for natural language processing (NLP). 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 [27] and have fast implementations [15]. They continue to give better performance on many downstream supervised tasks such as IMDB sentiment classification [18] than purely unsupervised LSTM representations [17, 13, 21]. Meanwhile there is evidence suggesting that simpler *linear* schemes give compact representations that provide most of the benefits of word-level LSTM embeddings [28, 3]. These linear schemes consist of simply adding up, with a few modifications, standard pretrained word embeddings such as GloVe or word2vec [20, 23].

Results have shown the learnability of linear classifiers of samples compressed via linear compression matrices satisfying certain strong compressed sensing properties [7]. Similar theory has also been used to show how LSTM representations are at least as good as Bag-of-n-Grams for linear classification, up to a dimension-dependent approximation error; in this settings the LSTM can be seen as also computing a compressed BonG representation, though in lower memory [1]. However, these results depend on the encoding of words using i.i.d. random vectors (e.g. Rademacher or Gaussian) in order to preserve BonG information; in practice, NLP practitioners commonly use pretrained word embeddings, especially for linear classification tasks, where they perform much better in practice. We will discuss what can be done in a setting such as this one, where powerful theories of compressed sensing/sparse recovery [8] may be more difficulty to apply. In doing so we make the following contributions:

1. We present the empirical finding that using pretrained embeddings (GloVe / word2vec) instead of random vectors improves the ability to preserve Bag-of-Words (BoW) information, i.e. they are better for sensing BoW signals. This finding is surprising as such embeddings do not satisfy standard compressed sensing properties that guarantee recovery, and indeed their training objectives seem to contradict our intuitions about what vectors are good for sensing, even when restricting to certain sparse signal distributions.

2. We motivate some theoretical justification for this surprising finding using a new sparse recovery property characterizing when nonnegative signals can be reconstructed by ℓ_1-minimization using a geometric result in compressed sensing. Unlike many guarantees for sparse recovery, whether local or global, this condition can be efficiently verified for a given matrix and signal support.

Most of this thesis is based on joint work with Sanjeev Arora, Nikunj Saunshi, and Kiran Vodrahalli published in the Proceedings of the 6th International Conference on Learning Representations (ICLR 2018) [1].

Chapter 2

Related Work

2.1 Compressed Bag-of-n-Grams Representations

Representations of BonG vectors have been studied through the lens of compression by [22], who computed representations based on classical lossless compression algorithms using a linear program (LP). 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. [21] 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.

2.2 The Sparse Recovery Problem

The novelty in the current paper is the connection to compressed sensing, which is concerned with recovering high-dimensional sparse signals $x \in \mathbb{R}^N$ from low-dimensional linear measurements Ax, specifically by studying conditions on matrix $A \in \mathbb{R}^{d\times N}$

when this is possible. In the noiseless case this is formulated as

$$\text{minimize} \quad \|w\|_0 \quad \text{subject to} \quad Aw = z \tag{2.1}$$

where $A \in \mathbb{R}^{d \times N}$ is the *design matrix* and $z = Ax$ is the *measurement vector*. Since ℓ_0-minimization is NP-hard, a foundational approach is to use its convex surrogate, the ℓ_1-norm, and characterize when the solution to (2.1) is equivalent to that of the following LP, known as *basis pursuit* (BP):

$$\text{minimize} \quad \|w\|_1 \quad \text{subject to} \quad Aw = z \tag{2.2}$$

Related approaches such as *Basis Pursuit Denoising* (LASSO) and the *Dantzig Selector* generalize BP to handle signal or measurement noise [12]; however, the word embeddings case is noiseless so these methods reduce to BP. Note that throughout this work we will say that an ℓ_1-minimization method *recovers* x from Ax if its optimal solution is unique and equivalent to the optimal solution of (2.1).

An alternative way to approximately solve (2.1) is to use a greedy algorithm such as *matching pursuit* (MP) or *orthogonal matching pursuit* (OMP), which pick basis vectors one at a time by multiplying the measurement vector by A^T and choosing the column with the largest inner product [26].

2.3 Guaranteeing Perfect Recovery

One condition through which recovery can be guaranteed is the *Restricted Isometry Property* (RIP):

Definition 2.3.1. *$A \in \mathbb{R}^{d \times N}$ is (k, ε)-RIP if for all k-sparse $x \in \mathbb{R}^N$*

$$(1 - \varepsilon)\|x\|_2 \leq \|Ax\|_2 \leq (1 + \varepsilon)\|x\|_2$$

A line of work started by [8] used the RIP property to characterize matrices A such that (2.1) and (2.2) have the same minimizer for any k-sparse signal x; this occurs with overwhelming probability when $d = \Omega\left(k \log \frac{N}{k}\right)$ and $\sqrt{d}A_{ij} \sim \mathcal{N}(0,1)\ \forall\ i,j$ or $\sqrt{d}A_{ij} \sim \mathcal{U}\{-1,1\}\ \forall\ i,j$.

Since the ability to recover a signal x from a representation Ax implies information preservation, a natural next step is to consider learning after compression. [7] show that for m i.i.d. k-sparse samples $\{(x_i, y_i)\}_{i=1}^m$ and a $(2k, \varepsilon)$-RIP matrix A, the hinge loss of a classifier trained on $\{(Ax_i, y_i)\}_{i=1}^m$ is bounded by that of the best linear classifier over the original samples. Theorem 3.1.1 provides a generalization of this result to any convex Lipschitz loss function.

RIP is a strong requirement, both because it is not necessary for perfect, stable recovery of k-sparse vectors using $\tilde{\mathcal{O}}(k)$ measurements and because in certain settings we are interested in using the above ideas to recover specific signals — those statistically likely to occur—rather than all k-sparse signals. The usual necessary and sufficient condition to recover any vector $x \in \mathbb{R}^N$ with index support set $S \subset [N]$ is the *local nullspace property* (NSP), which is implied by RIP:

Definition 2.3.2 ([12])**.** *A matrix $A \in \mathbb{R}^{d\times N}$ satisfies NSP for a set $S \subset [N]$ if $\|w_S\|_1 < \|w_{\overline{S}}\|_1$ for all nonzero $w \in \ker(A) = \{v : Av = \mathbf{0}_d\}$.*

Theorem 2.3.1 ([12])**.** *BP (2.2) recovers any $x \in \mathbb{R}_+^N$ with $\operatorname{supp}(x) = S$ from Ax iff A satisfies NSP for S.*

A related condition that implies NSP is the *local restricted eigenvalue property* (REP):

Definition 2.3.3 ([24])**.** *A matrix $A \in \mathbb{R}^{d\times N}$ satisfies γ-REP for a set $S \subset [N]$ if $\|Aw\|_2 \geq \gamma\sqrt{d}\|w\|_2$ whenever $\|w_{\overline{S}}\|_1 \leq \|w_S\|_1$.*

Lastly, a simple condition that can sometimes provide recovery guarantees is *mutual incoherence*:

Definition 2.3.4. *$A \in \mathbb{R}^{d \times N}$ is μ-incoherent if $\max_{a,a'} |a^T a'| \leq \mu$, where the maximum is taken over any two distinct columns a, a' of A.*

While incoherence is easy to verify (unlike the previous recovery properties), word embeddings tend to have high coherence due to the training objective pushing together vectors of co-occurring words.

Chapter 3

Word Embeddings and Compressed Sensing

In this section we discuss the connection between compressed sensing, sparse language representations, and word embeddings. We first examine the compressed learning setting of [7], whose work and subsequent results require an RIP property, which is only efficiently satisfied by random vectors. However, while the word embedding matrix arguably do not satisfy such a condition, we then show how they are surprisingly good sensing vectors, in terms of the low-dimensionality they need to recover BoW signals using ℓ_1-minimization. This property is further shown to be dependent on the language distribution used to train the embeddings.

3.1 The Connection to Compressed Learning

Following the early breakthroughs in compressed sensing, [7] studied whether it is possible to use its low-dimensional output as a surrogate representation for classification. Their result, a learning-theoretic bound on the loss of an SVM classifier in the compressed domain compared to the best classifier in the original domain, was further generalized to handle Lipschitz losses over arbitrary sets [1]:

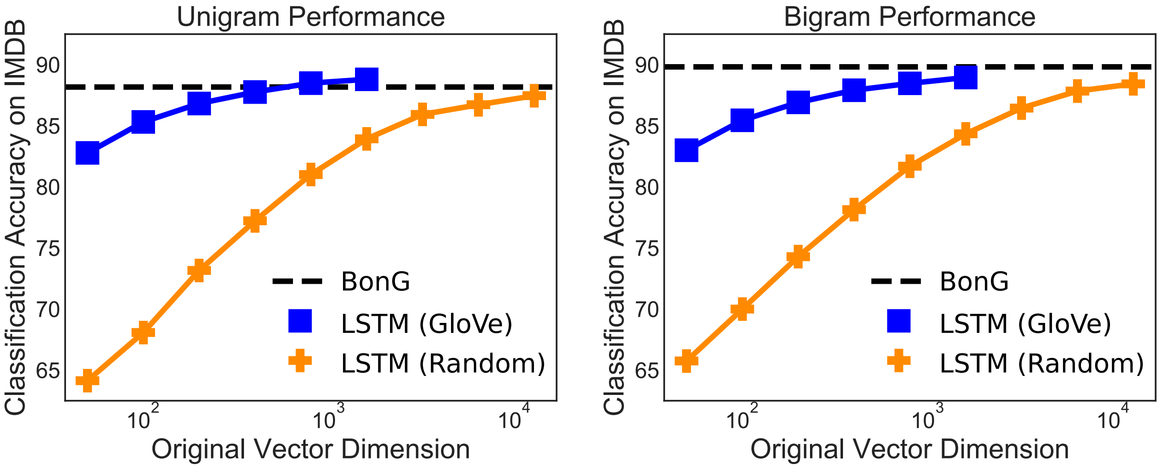

Figure 3.1: IMDB performance of unigram (left) and bigram (right) linear LSTM embeddings compared to the original word embedding dimension.

Theorem 3.1.1 ([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*

$$\ell_{\mathcal{D}}(\hat{w}_A) \leq \ell_{\mathcal{D}}(w_0) + \mathcal{O}\left(\lambda R\|w_0\|_2\sqrt{\varepsilon + \frac{1}{m}\log\frac{1}{\delta}}\right) \tag{3.1}$$

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

The result follows from an analysis of 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, together with standard statistical learning arguments for regularized linear classifiers.

The application of this theorem to study the simplest linear schemes – BoW vectors compressed as sums of word embeddings – directly follows if one uses standard i.i.d. random ensembles for A. [1] further extend the result to an existence statement about LSTMs – that there exists one with hidden dimension (memory) of size $\tilde{\mathcal{O}}\left(\frac{nT}{\varepsilon^2}\right)$

that can compute a compression of a BonG vector of any document of length at most T such that the compression is also RIP. Together with Theorem 3.1.1 this implies that representations computed by such an LSTM are at least as powerful, up to an approximation error that decreases in d, as BonGs for linear classification.

The results of [7, 1] depend heavily on the RIP properties of the compression matrix A, which requires words to be represented by random d-dimensional embeddings. In practice, however, NLP tasks are often solved us *word embeddings* - fixed vectors such as word2vec [20] or GloVe [23] which are trained such that more similar words have a higher cosine similarity, where similarity is defined as some function of how often pairs of words occur together within a fixed window. Because word embeddings may be highly coherent (e.g. synonyms) there exist k-sparse vectors for which the embedding matrix will not preserve the norm within a reasonable distortion, and so embeddings do not satisfy RIP by virtue of their objective.

Nevertheless, we see in Figure 3.1 that pretrained word embeddings have much better performance as inputs to linear representation schemes (here the task is the IMDB classification task [18] and vectors are trained on a large corpus of Amazon reviews [19]) In fact, while representing documents as sums of random vectors causes performance to increase much as Theorem 3.1.1 predicts – asymptotically approaching BoW performance – word embeddings quickly match and even surpass it in the unigram case. Based on the same intuitions that motivated their introduction in the first place, it might make sense that word embeddings exhibit these superior properties; the corpus information they encode allows better generalization. However, the same result challenges the compressed sensing view of [1] – that in reality LSTM representations are computed by vectors that do not satisfy nice sparse recovery properties, and so their results do not extend to such practical settings. Answering this question motivates the remainder of our work.

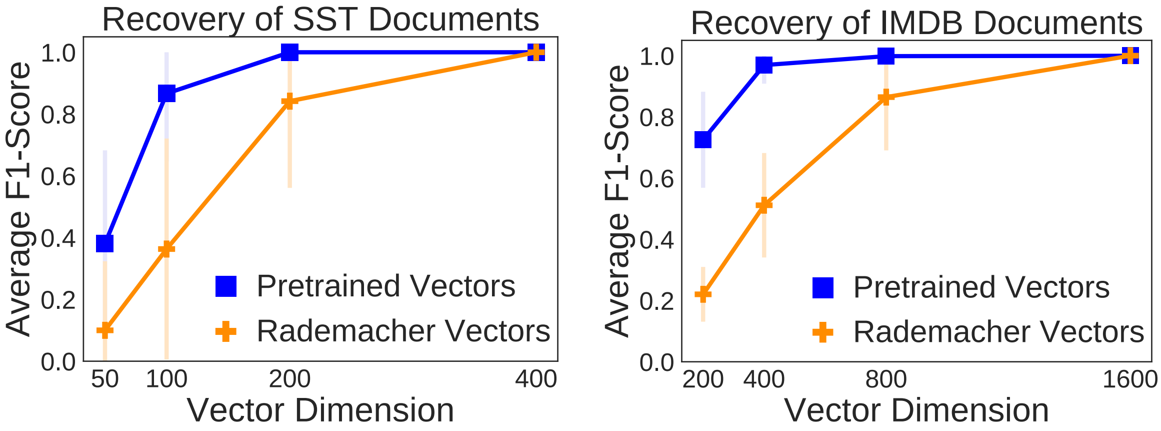

Figure 3.2: Average F_1-score of 200 recovered BoW vectors from SST (left) and IMDB (right) compared to dimension. Pretrained word embeddings (SN trained on Amazon reviews) need half the dimensionality of normalized Rademacher vectors to achieve near-perfect recovery. Note that IMDB documents are on average more than ten times longer than SST documents.

3.2 The Surprising Efficiency of Word Embeddings for Sparse Recovery

In recent years word embeddings have been discovered to have many remarkable properties, most famously the ability to solve analogies [20]. The connection made by [1] to compressed sensing indicates that they should have another: preservation of sparse signals as low-dimensional linear measurements. To examine this we subsample documents from the SST [25] and IMDB [18] classification datasets, embed them as d-dimensional unigram embeddings $z = Ax$ for $d = 50, 100, 200, \dots, 1600$ (where $A \in \mathbb{R}^{d \times V}$ is the matrix of word embeddings and x is a document's BoW vector), solve the following LP, known as *Basis Pursuit* (BP), which is the standard ℓ_1-minimization problem for sparse recovery in the noiseless case:

$$\text{minimize} \quad \|w\|_1 \quad \text{subject to} \quad Aw = z \tag{3.2}$$

Success is measured as the F_1 score of retrieved words. We use Squared Norm (SN) vectors [2] trained on a corpus of Amazon reviews [19] and normalized i.i.d.

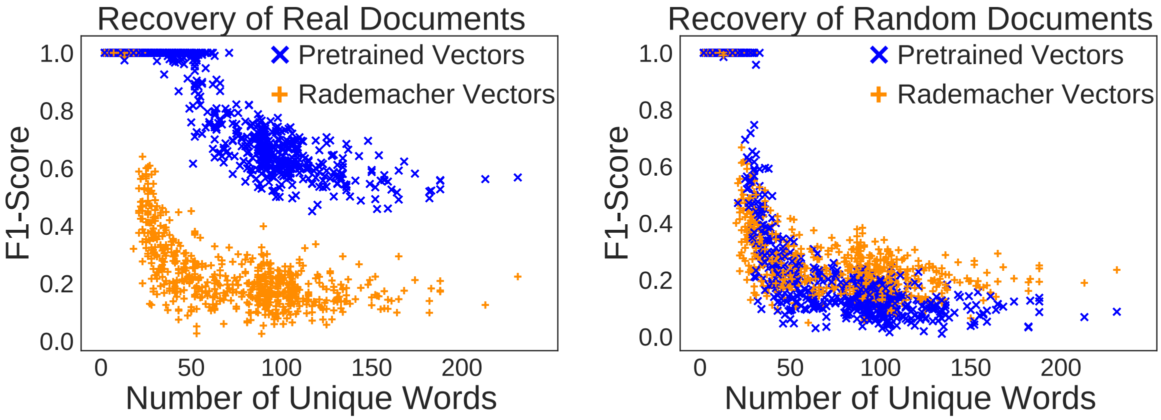

Figure 3.3: F_1-score of 1000 recovered BoWs compared to number of unique words. Real documents (left) are drawn from the SST and IMDB corpora; random signals (right) are created by picking words at random. For $d = 200$, pretrained embeddings are better than Rademacher vectors as sensing vectors for natural language BoW but are worse for random sparse signals.

Rademacher vectors as a baseline. SN is used due to similarity to GloVe and its formulation via an easy-to-analyze generative model that may provide a framework to understand the results, while the Amazon corpus is used for its semantic closeness to the sentiment datasets.

Figure 3.2 and 3.3 show that pretrained embeddings require a lower dimension d than random vectors to recover natural language BoW. This is surprising as the training objective goes against standard conditions such as approximate isometry and incoherence; indeed as shown in Figure 3.3 recovery is poor for randomly generated word collections. The latter outcome indicates that the fact that a document is a set of mutually meaningful words is important for sparse recovery using embeddings trained on co-occurrences. We achieve similar results with other objectives (e.g. GloVe/word2vec) and other corpora, although from Figure 3.4 we see that SN vectors are most efficient and the only embeddings where normalizing is not needed for good performance. We also see some sensitivity to the recovery method, as ℓ_1-minimization methods work well but greedy methods, such as Orthogonal Matching Pursuit (OMP), sometimes work poorly, likely due to their dependence on incoherence [26].

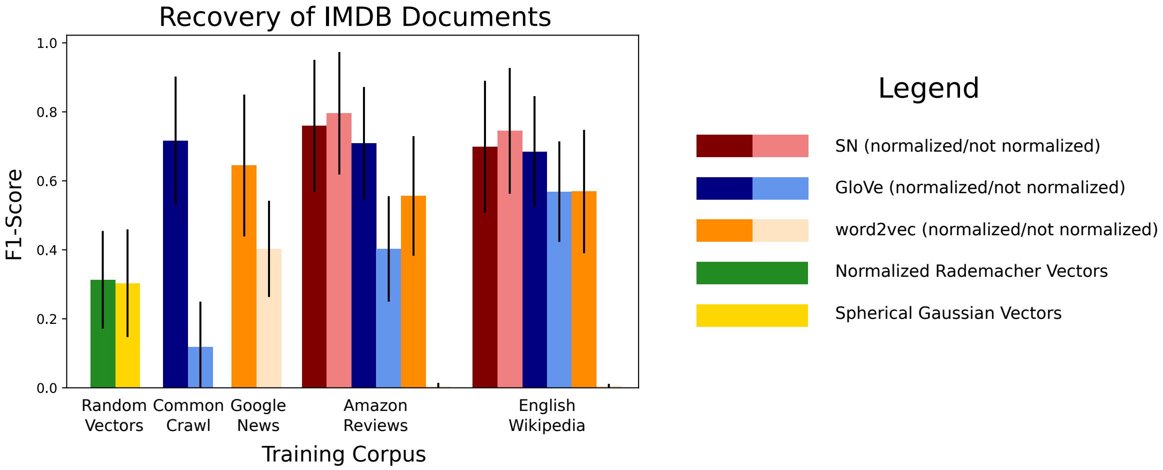

Figure 3.4: Efficiency of pretrained embeddings as sensing vectors at $d = 300$ dimensions, measured via the F_1-score of the original BoW. 200 documents from each dataset were compressed and recovered in this experiment. For fairness, the number of words V is the same for all embeddings so all documents are required to be subsets of the vocabulary of all corpora. word2vec embeddings trained on Google News and GloVe vectors trained on Common Crawl were obtained from public repositories [20, 23] while Amazon and Wikipedia embeddings were trained for 100 iterations using a symmetric window of size 10, a min count of 100, for SN/GloVe a cooccurrence cutoff of 1000, and for word2vec a down-sampling frequency cutoff of 10^{-5} and a negative example setting of 3. 300-dimensional normalized random vectors are used as a baseline.

Chapter 4

A Geometric Understanding of Bag-of-Words Recovery

As shown in Figure 3.3, the success of pretrained embeddings for linear sensing is a local phenomenon; recovery is only efficient for naturally occurring collections of words. However, applying statistical RIP/incoherence ideas [4] to explain this is ruled out since they require collections to be incoherent with high probability, whereas word embeddings are trained to give high inner product to words appearing together. Thus an explanation must come from some other, weaker condition. The usual necessary and sufficient requirement for recovering all signals with support $S \subset [N]$ is the *local nullspace property* (NSP), which stipulates that vectors in the kernel of A not have too much mass on S (see Definition 2.3.2). While NSP and related properties such as *restricted eigenvalue* (see Definition 2.3.3) are hard to check, we can impose some additional structure to formulate an intuitive, verifiable perfect recovery condition for our setting.

4.1 Nonnegative Signal Recovery

Apart from incoherence, recovery properties are often hard to show empirically. However, we are compressing BoW vectors, so our signals are nonnegative and we can impose an additional constraint on (2.2):

$$\text{minimize} \quad \|w\|_1 \quad \text{subject to} \quad Aw = z, \quad w \geq \mathbf{0}_d \tag{4.1}$$

The following geometric result provides guarantees for this *nonnegative basis pursuit* (BP+) problem:

Theorem 4.1.1 ([10])**.** *Consider a matrix $A \in \mathbb{R}^{d \times N}$ and an index subset $S \subset [N]$ of size k. Then any nonnegative vector $x \in \mathbb{R}_+^N$ with support* $\operatorname{supp}(x) = S$ *is recovered from Ax by BP+* (4.1) *iff the columns of A indexed by S comprise the vertices of a k-dimensional face of the convex hull* $\operatorname{conv}(A)$ *of the columns of A together with the origin.*

The polytope condition is equivalent to *nonnegative NSP* (NSP+), a weaker form of NSP:

Definition 4.1.1 ([11])**.** *A matrix $A \in \mathbb{R}^{d \times N}$ satisfies NSP+ for a set $S \subset [N]$ if $w_{\overline{S}} \geq \mathbf{0}_N \implies \sum\limits_{i=1}^{N} w_i > 0$ for all nonzero $w \in \ker(A)$.*

Lemma 4.1.1. *If $A \in \mathbb{R}^{d \times N}$ satisfies NSP for some $S \subset [N]$ then it also satisfies NSP+ for S.*

Proof (Adapted from [11]). Since A satisfies NSP, we have $\|w_S\|_1 < \|w_{\overline{S}}\|_1$. Then for a nonzero $w \in \ker(A)$ such that $w_{\overline{S}} \geq \mathbf{0}$ we will have

$$\sum_{i=1}^{N} w_i = \sum_{i \in S} w_i + \sum_{j \in \overline{S}} w_j \geq -\sum_{i \in S} |w_i| + \sum_{j \in \overline{S}} |w_j| = -\|w_S\|_1 + \|w_{\overline{S}}\|_1 > 0$$

□

Lemma 4.1.2. *BP+ recovers any $x \in \mathbb{R}^N_+$ with $\operatorname{supp}(x) = S$ from Ax iff A satisfies NSP+ for S.*

Proof. ($\implies$): For any nonzero $w \in \ker(A)$ such that $w_{\overline{S}} \geq \mathbf{0}$, $\exists\ \lambda > 0$ such that $x + \lambda w \geq \mathbf{0}_N$ and $A(x + \lambda w) = Ax$. Since BP+ uniquely recovers x, we have $\|x + \lambda w\|_1 > \|x\|_1$, so NSP+ follows from the following inequality and the fact that λ is positive:

$$0 < \|x + \lambda w\|_1 - \|x\|_1 = \sum_{i=1}^{N} (x_i + \lambda w_i) - \sum_{i=1}^{N} x_i = \lambda \sum_{i=1}^{N} w_i$$

$$\implies \sum_{i=1}^{N} w_i > 0$$

($\impliedby$): For any $x' \geq \mathbf{0}$ such that $Ax' = Ax$ we have that $w = x' - x \in \ker(A)$ and $w_{\overline{S}} = x'_{\overline{S}} \geq \mathbf{0}$ since the support of x is S. Thus by NSP+ we have that $\sum\limits_{i=1}^{N} w_i > 0$, which yields

$$\|x'\|_1 - \|x\|_1 = \sum_{i=1}^{N} x'_i - \sum_{i=1}^{N} x_i = \sum_{i=1}^{N} w_i > 0$$

Thus BP+ will recover x uniquely. □

Lemma 4.1.2 shows that NSP+ is equivalent to the polytope condition in Theorem 4.1.1, as they are both necessary and sufficient conditions for BP+ recovery.

4.2 The Supporting Hyperplane Property

Theorem 4.1.1 equates perfect recovery of a BoW vector via BP+ with the vectors of its words being the vertices of some face of the polytope $\operatorname{conv}(A)$. The property holds for incoherent columns since the vectors are far enough that no one vector is inside the simplex formed by any k others. On the other hand, pretrained embeddings satisfy it by having commonly co-occurring words close together and other words far away, making it easier to form a face from columns indexed by the support of a BoW. We formalize this intuition as the *Supporting Hyperplane Property* (SHP):

Definition 4.2.1. *A matrix $A \in \mathbb{R}^{d\times N}$ satisfies S-SHP for subset $S \subset [N]$ if its columns are in general position and there is a hyperplane containing the set A_S of columns of A indexed by S such that the set of all other columns of A together with the origin are on one side of the hyperplane.*

4.2.1 Characterizing Nonnegative Sparse Recovery

We now show that the SHP hyperplane is the *supporting hyperplane* of the face of $\operatorname{conv}(A)$ with vertices A_S, from which it follows by Theorem 4.1.1 that SHP *characterizes* recovery using BP+:

Theorem 4.2.1. *BP+ recovers any $x \in \mathbb{R}^N_+$ with $\operatorname{supp}(x) = S$ from Ax iff A satisfies S-SHP.*

Proof. By Theorem 4.1.1 it suffices to show equivalence of A being S-SHP with the columns A_S forming the vertices of a k-dimensional face of $\operatorname{conv}(A)$, where we can abuse notation to set $A \in \mathbb{R}^{d\times(N+1)}$, with the extra column being the origin $\mathbf{0}_d$, so long as we constrain $N+1 \notin S$. A *face* F of polytope P is defined as its intersection with some hyperplane such that all points in $P\backslash F$ lie on one side of the hyperplane.

($\implies$) Let F be the face of $\operatorname{conv}(A)$ formed by the columns A_S. Then there must be a supporting hyperplane H containing F. Since the columns of A are in general position, all columns $A_{\overline{S}} = A\backslash A_S$ lie in $\operatorname{conv}(A)\backslash F$ and hence must all be on one side of H, so H is the desired hyperplane.

($\Longleftarrow$): A subset $F \subset \mathbb{R}^d$ is a face of conv(A) if for some hyperplane $H = \{v : a^T v - b = 0\}$ we have $F = \text{conv}(A) \cap H$ and $\text{conv}(A) \backslash F \subseteq H_- = \{v : a^T v - b < 0\}$, where H_- is the negative halfspace of H. Define the simplex

$$\Delta_m = \{\lambda \in [0,1]^m : \sum_{i=1}^{m} \lambda_i = 1\}$$

Since A is S-SHP we have a hyperplane $H = \{v : a^T v - b = 0\}$ containing the columns A_S such that $A_{\overline{S}} \subset H_-$. Thus $a^T A_i - b = 0 \;\forall\; i \in S$ and $a^T A_i - b < 0 \;\forall\; i \notin S$. We also know that $F = \{\sum_{i \in S} \lambda_i A_i : \lambda \in \Delta_{|S|}\} \subseteq H$ by convexity of H. Since any point $y \in \text{conv}(A) \backslash F$ can be written as $y = \sum_{i=1}^{N+1} \lambda_i A_i$ for some $\lambda \in \Delta_{N+1}$ such that $\exists\; j \notin S$ such that $\lambda_j \neq 0$, we have that

$$a^T y - b = \sum_{i \in S} \lambda_i (a^T A_i - b) + \sum_{j \notin S} \lambda_j (a^T A_j - b) = \sum_{j \notin S} \lambda_j (a^T A_j - b) < 0$$

This implies that $\text{conv}(A) \backslash F \subseteq H_-$ and $F = \text{conv}(A) \cap H$, so since the columns of A are in general position F is a k-dimensional face of conv(A) whose vertices are the columns A_S. □

Thus perfect recovery of a BoW via BP+ is equivalent to the existence of a hyperplane separating embeddings of words in the document from those of the rest of the vocabulary. Together with Lemmas 4.1.1 and 4.1.2 Theorem 4.2.1 also shows that SHP is a weaker condition than the well-known nullspace property (NSP):

Corollary 4.2.1. *If a matrix $A \in \mathbb{R}^{d \times N}$ with columns in general position satisfies NSP for some $S \subset [N]$ then it also satisfies S-SHP.*

4.2.2 Verifying the Supporting Hyperplane Property

Recall that a matrix $\mathbb{R}^{d\times N}$ satisfies S-SHP for $S \subset [N]$ if there is a hyperplane containing the set of all columns of A indexed by S and the set of all other columns together with the origin are on one side of it. Due to Theorem 4.2.1, checking S-SHP allows us to know whether all nonnegative signals with index support S will be recovered by BP+ without actually running the optimization on any one of them.

To see that this property can be checked efficiently (that is, in time polynomial in the dimensions of A), we can consider the following feasibility problem over $h \in \mathbb{R}^{d+1}$:

$$\tilde{A}_i^T h = 0 \;\forall\; i \in S \qquad \text{and} \qquad \tilde{A}_i^T h + \varepsilon \leq 0 \;\forall\; i \notin S$$

$$\text{where} \quad \tilde{A} = \begin{pmatrix} A & \mathbf{0}_d \\ \mathbf{1}_N^T & 1 \end{pmatrix} \quad \text{and} \quad \varepsilon > 0$$

Here the equality constraint enforces the property that the hyperplane contains all support embeddings, while the inequality requires all non-support columns to be on the same side of the hyperplane as the origin. Since scaling h does not affect the constraint, if an h exists for any single $\varepsilon > 0$ it exists for all $\varepsilon > 0$. Therefore verifying S-SHP for a matrix A is equivalent to seeing if these constraints are feasible; because this can be determined by an LP, this shows that SHP is efficiently verifiable.

In practice such LPs are difficult to solve, so we can rewrite the optimization property into the following constrained convex problem (for $p \geq 1$)

$$\min_{h\in\mathbb{R}^{d+1}} \sum_{i\notin S} \max\left\{\tilde{A}_i^T h + \varepsilon, 0\right\}^p \quad \text{subject to} \quad \tilde{A}_S^T h = \mathbf{0}_{|S|}$$

In our experiments we set $\varepsilon = 1$ and $p = 3$ (to get a $\mathcal{C}^2$ objective) and adapt the second-order method from [6, Chapter 10]. Our implementation can be found at `https://github.com/NLPrinceton/sparse_recovery`.

4.3 Checking the Supporting Hyperplane Property for Word Embeddings

Intuitively, words in the same document are trained to have similar embeddings and so will be easier to separate out, providing some justification for why pretrained vectors are better for sensing. We verify that SHP is indeed more likely to be satisfied by such designs in Figure 4.1, which also serves as an empirical check of Theorem 4.2.1 since SHP satisfaction implies BP recovery as the latter can do no better than BP+. We further compare to recovery using OMP/OMP+ (the latter removes negative values and recomputes the set of atoms at each iteration); interestingly, while OMP+ recovers the correct signal from SN almost as often as BP/BP+, it performs quite poorly for GloVe, indicating that these embeddings may have quite different sensing properties despite similar training objectives.

As similarity properties that may explain these results also relate to downstream task performance, we conjecture a relationship between embeddings, recovery, and classification that may be understood under a generative model (see Section 4.4). However, the compressed learning bounds of [7, 1] depend on RIP, not recovery, so these experiments by themselves do not apply. They do show that the compressed sensing framework remains relevant even in the case of non-random, pretrained word embeddings.

4.4 Insights from a Generative Model

In the previous section we gave some intuition for why pretrained word embeddings are efficient sensing vectors for natural language BoW by examining a geometric characterization of local equivalence due to [10] in light of the usual similarity properties of word embeddings. However, this analysis does not provide a rigorous theory for

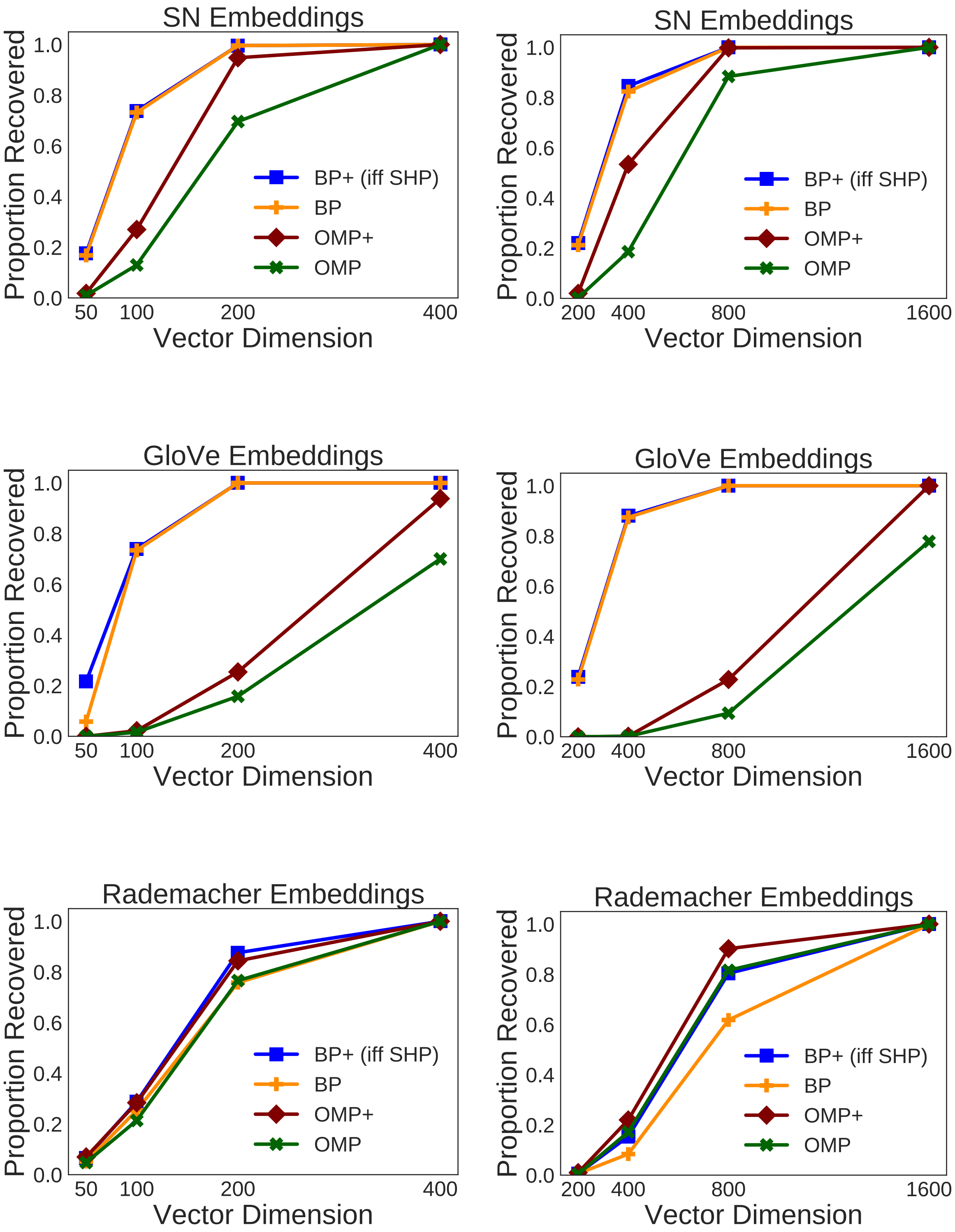

Figure 4.1: Proportion of 500 randomly sampled documents from SST (left) and IMDB (right) that are perfectly recovered from linear measurements.

our empirical results. In this section we briefly discuss a model-based justification that may lead to a stronger understanding.

We need a model relating BoW generation to the word embeddings trained over words co-occurring in the same BoW. As a starting point consider the model of [2], in which a corpus is generated by a random walk c_t over the surface of a ball in $\mathbb{R}^d$; at each t a word w is emitted w.p.

$$\mathbb{P}(w|c_t) \propto \exp\langle c_t, v_w \rangle \tag{4.2}$$

Minimizing the SN objective approximately maximizes the corpus likelihood.

Thus in an approximate sense a document of length T is generated by setting a *context vector* c and emitting T words via (4.2) with $c_t = c$. This model is a convenient one for analysis due its simplicity as well as the fact that the approximate maximum likelihood document vector is the sum of the embeddings of words in the document. Building upon the intuition established following Theorem 4.2.1 one can argue that, if we have the true latent SN vectors, then embeddings of words in the same document (i.e. emitted by the same context vector) will be close to each other and thus easy to separate from the embeddings of other words.

However, we find empirically that not all of the T words closest to the sum of the word embeddings (i.e. the context vector) are the ones emitted; indeed individual word vectors in a document may have small, even negative inner product with the context vector and still be recovered via BP. Thus any further theoretical argument must also be able to handle the recovery of lower probability words whose vectors are further away from the context vector than those of words that do not appear in the document. We thus leave to future work the challenge of explaining why embeddings resulting from this (or another) model provide such efficient sensing matrices for natural language BoW.

Chapter 5

Conclusion

We have demonstrated and analyzed a surprising new property of distributed word embeddings: that they form more efficient sensing matrices for natural language Bag-of-Words vectors. In an effort to understand this surprising finding, we have further provided a new characterization of perfect recovery using nonnegative basis pursuit via the Supporting Hyperplane Property (SHP), which we show is also efficiently verifiable via an LP. Using this understanding and experimental analysis of the recovery behavior under different signal types, we proposed an explanation for our observations, although a full rigorous understanding is left to future work.

Though motivated by the problem of compressed learning [7, 1], our results demonstrate only the recovery properties of word embeddings, and so an important direction for future work is to find intermediate properties that both guarantee recovery and provide bounds on the loss in the compressed domain. The information preservation and recovery properties exhibited by these word embeddings may also have many interesting NLP applications, especially if similar sensing properties can be induced in n-gram embeddings. Besides improving classification performance [16], such representations may also point to simple approaches for NLP settings with low-dimensional encoding or decoding, such machine translation or language generation.

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] Alexander Barg, Arya Mazumdar, and Rongrong Wang. Restricted isometry property of random subdictionaries. *IEEE Transactions on Information Theory*, 61, 2015.

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

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

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

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

[9] 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.

[10] David L. Donoho and Jared Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences of the United States of America.*, 102:9446–9451, 2005.

[11] Simon Foucart and David Koslicki. Sparse recovery by means of nonnegative least squares. *IEEE Signal Processing Letters*, 21:498–502, 2014.

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

[13] 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.

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

[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] 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 *To Appear in the Proceedings of the 56th Annual Meeting of the ACL*, 2018.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the North American Chapter of the ACL*, 2018.

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

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

[24] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research*, 11(Aug):2241–2259, 2010.

[25] 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.

[26] Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50, 2004.

[27] 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.

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