Sublinear Time Approximation of Text Similarity Matrices

Archan Ray, Nicholas Monath†, Andrew McCallum, Cameron Musco

College of Information and Computer Sciences, University of Massachusetts Amherst
{ray, nmonath, mccallum, cmusco}@cs.umass.edu

Abstract

We study algorithms for approximating pairwise similarity matrices that arise in natural language processing. Generally, computing a similarity matrix for n data points requires $\Omega(n^2)$ similarity computations. This quadratic scaling is a significant bottleneck, especially when similarities are computed via expensive functions, e.g., via transformer models. Approximation methods reduce this quadratic complexity, often by using a small subset of exactly computed similarities to approximate the remainder of the complete pairwise similarity matrix.

Significant work focuses on the efficient approximation of positive semidefinite (PSD) similarity matrices, which arise e.g., in kernel methods. However, much less is understood about indefinite (non-PSD) similarity matrices, which often arise in NLP. Motivated by the observation that many of these matrices are still somewhat close to PSD, we introduce a generalization of the popular *Nyström method* to the indefinite setting. Our algorithm can be applied to any similarity matrix and runs in sublinear time in the size of the matrix, producing a rank-s approximation with just $O(ns)$ similarity computations.

We show that our method, along with a simple variant of CUR decomposition, performs very well in approximating a variety of similarity matrices arising in NLP tasks. We demonstrate high accuracy of the approximated similarity matrices in the downstream tasks of document classification, sentence similarity, and cross-document coreference.

1 Introduction

Many machine learning tasks center around the computation of pairwise similarities between data points using an appropriately chosen similarity function. E.g., in kernel methods, a non-linear kernel inner product is used to measure similarity, and often to construct a pairwise kernel similarity matrix. In natural language processing, document or sentence similarity functions (e.g., cross-encoder transformer models (Devlin et al. 2018) or word mover's distance (Piccoli and Rossi 2014; Kusner et al. 2015))) are key components of cross-document coreference (Cattan et al. 2020) and passage retrieval for question answering (Karpukhin et al. 2020). String-similarity functions are used to model name aliases (Tam et al. 2019) and for morphology (Rastogi, Cotterell, and Eisner 2016).

† Now at Google.

Computing all pairwise similarities for a data set with n points requires $\Omega(n^2)$ similarity computations. This can be a major runtime bottleneck, especially when each computation requires the evaluation of a neural network or other expensive operation. One approach to avoid this bottleneck is to produce a compressed approximation to the $n \times n$ pairwise similarity matrix $\mathbf{K}$ for the data set, but avoid ever fully forming this matrix and run in sub-quadratic time (i.e., with running time less than $O(n^2)$, or sublinear in the size of $\mathbf{K}$). The compressed approximation, $\tilde{\mathbf{K}}$, can be used in place of $\mathbf{K}$ to quickly access approximate pairwise similarities, and in methods for near neighbor search, clustering, and regression, which would typically involve $\mathbf{K}$.

1.1 Existing Methods

Similarity matrix approximation is very well-studied, especially in the context of accelerating kernel methods and Gaussian process regression. Here, $\mathbf{K}$ is typically positive semidefinite (PSD). This structure is leveraged by techniques like the random Fourier features and Nyström methods (Rahimi and Recht 2007; Le, Sarlós, and Smola 2013; Williams and Seeger 2001; Yang et al. 2012), which approximate $\mathbf{K}$ via a rank-s approximation $\tilde{\mathbf{K}} = \mathbf{Z}\mathbf{Z}^T$, for $s \ll n$ and $\mathbf{Z} \in \mathbb{R}^{n \times s}$. These methods have runtimes scaling linearly in n and sublinear in the matrix size. They have been very successful in practice (Huang et al. 2014; Meanti et al. 2020), and often come with strong theoretical bounds (Gittens and Mahoney 2016; Musco and Musco 2017; Musco and Woodruff 2017).

Unfortunately, most similarity matrices arising in natural language processing, such as those based on cross-encoder transformers (Devlin et al. 2018) or word mover's distance (Piccoli and Rossi 2014), are *indefinite* (i.e., non-PSD). For such matrices, much less is known. Sublinear time methods have been studied for certain classes of similarities (Bakshi and Woodruff 2018; Oglic and Gärtner 2019; Indyk et al. 2019), but do not apply more generally. Classic techniques like low-rank approximation via the SVD or fast low-rank approximation via random sketching (Frieze, Kannan, and Vempala 2004; Sarlos 2006; Drineas, Mahoney, and Muthukrishnan 2008) generally must form all of $\mathbf{K}$ to approximate it, and so run in $\Omega(n^2)$ time. There are generic sublinear time sampling methods, like CUR decomposition (Drineas, Kannan, and Mahoney 2006; Wang, Zhang, and Zhang 2016), which are closely related to Nyström approximation. How-

ever, as we will see, the performance of these methods varies greatly depending on the application.

1.2 Our Contributions

Algorithmic. Our first contribution is a simple variant of the Nyström method that applies to symmetric indefinite similarity matrices[1]. The Nyström method (Williams and Seeger 2001) approximates a PSD similarity matrix $\mathbf{K}$ by sampling a set of $s \ll n$ *landmark points* from the dataset, computing their similarities with all other points (requiring $O(ns)$ similarity computations), and then using this sampled set of similarities to reconstruct all of $\mathbf{K}$. See Sec. 2.

Our algorithm is motivated by the observation that many indefinite similarity matrices arising in NLP are *somewhat close to PSD* – they have relatively few negative eigenvalues. Thus, a natural approach would be simply to apply Nyström to them. However, even for matrices with just a few small negative eigenvalues, this fails completely. We instead show how to 'minimally correct' our matrix to be closer to PSD, before applying Nyström. Specifically, we apply an eigenvalue shift based on the minimum eigenvalue of a small random principal submatrix of $\mathbf{K}$. We call our method *Submatrix-Shifted Nyström*, or *SMS-Nyström*. SMS-Nyström is extremely efficient, and, while we do not give rigorous approximation bounds, it recovers the strong performance of the Nyström method on many near PSD-matrices.

Empirical. Our second contribution is a systematic evaluation of a number of sublinear time matrix approximation methods in NLP applications. We consider three applications involving indefinite similarity matrices: 1) computing document embeddings using word mover's distance (Kusner et al. 2015), for four different text classification tasks; 2) approximating similarity matrices generated using cross-encoder BERT (Devlin et al. 2018) and then comparing performance in three GLUE tasks: STS-B (Cer et al. 2017), MRPC (Dolan and Brockett 2005) and RTE (Bentivogli et al. 2009), which require predicting similarity, semantic equivalence, and entailment between sentences; 3) approximating the similarity function used to determine coreference relationships across documents in a corpus of news articles mentioning entities and events (Cybulska and Vossen 2014; Cattan et al. 2020).

We show that both SMS-Nyström, and a simple variant of CUR decomposition yield accurate approximations that maintain downstream task performance in all these tasks while greatly reducing the time and space required as compared to the exact similarity matrix. They typically significantly outperform the classic Nyström method and other CUR variants.

1.3 Other Related Work

Our work fits into a vast literature on randomized methods for matrix approximation (Mahoney 2011; Woodruff et al. 2014). There is significant work on different sampling distributions and theoretical bounds for both the Nyström and CUR methods (Goreinov, Tyrtyshnikov, and Zamarashkin 1997; Drineas, Mahoney, and Cristianini 2005; Drineas, Mahoney, and Muthukrishnan 2008; Zhang, Tsang, and Kwok 2008; Kumar, Mohri, and Talwalkar 2012; Wang and Zhang 2013; Talwalkar and Rostamizadeh 2014). However, more advanced methods generally require reading all of $\mathbf{K}$ and so do not avoid $\Omega(n^2)$ time. In fact, any method with non-trivial worst-case guarantees on general matrices cannot run less than $O(n^2)$ time. If the entire mass of the matrix is placed on a single entry, all entries must be accessed to find it.

A number of works apply Nyström variants to indefinite matrices. Belongie et al. (2002) show that the Nyström method can be effectively applied to eigenvector approximation for indefinite matrices, specifically in application to spectral partitioning. However, they do not investigate the behavior of the method in approximating the similarity matrix itself. Gisbrecht and Schleif (2015) shows that, in principal, the classic Nyström approximation converges to the true matrix when the similarity function is continuous over $\mathbb{R}$. However, we observe poor finite sample performance of this method on text similarity matrices. Other work exploits assumptions on the input points – e.g. that they lie in a small number of labeled classes, or in a low-dimensional space where distances correlate with the similarity (Schleif, Gisbrecht, and Tino 2018). This later assumption is made implictly in recent work on anchor-net based Nyström (Cai, Nagy, and Xi 2021), and while it may hold in many settings, in NLP applications, it is often not clear how to find such a low-dimensional representation. By removing the above assumptions, our work is well suited for applications in NLP, which often feed two inputs (e.g., sentences) into a neural network (e.g., transformer or MLP) to compute similarities.

There is also significant related work on modifying indefinite similarity matrices to be PSD, including via eigenvalue transformations and shifts (Chen, Gupta, and Recht 2009; Gisbrecht and Schleif 2015). These modifications would allow the matrix to be approximated with the classic Nyström method. However, this work does not focus on sublinear runtime, typically using modifications that require $\Omega(n^2)$ time.

Finally, outside of similarity matrix approximation, there are many methods that seek to reduce the cost of similarity computation. One approach is to reduce the number of similarity computations. Examples include locality sensitive hashing (Gionis et al. 1999; Lv et al. 2007), distance preserving embeddings (Hwang, Han, and Ahn 2012), and graph based algorithms (Orchard 1991; Dong, Moses, and Li 2011) for near-neighbor search. Another approach is to reduce the cost of each similarity computation, e.g., via model distillation for cross-encoder-based similarity (Sanh et al. 2019; Jiao et al. 2019; Michel, Levy, and Neubig 2019; Lan et al. 2019; Zafrir et al. 2019; Humeau et al. 2019). However, model distillation requires significant additional training time to fit the reduced model, unlike our proposed approach which requires only $O(ns)$ similarity computations. There is also work on random features methods and other alternatives to expensive similarity functions, such as those based on the word-movers distance (Cuturi 2013; Wu et al. 2018, 2019).

[1] While *asymmetric* similarity matrices do arise, we focus on the symmetric case. In our experiments, simply symmetrizing and then approximating these matrices yields good performance.

2 Submatrix-Shifted Nyström

In this section, we introduce the Nyström method for PSD matrix approximation, and describe our modification of this method for application to indefinite similarity matrices.

2.1 The Nyström Method

Let $\mathcal{X} = \{x_i\}_{i=1}^n$ be a dataset with n datapoints, $\Delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a similarity function, and $\mathbf{K} \in \mathbb{R}^{n \times n}$ be the corresponding similarity matrix with $\mathbf{K}_{ij} = \Delta(x_i, x_j)$.

The Nyström method samples s landmark points – let $\mathbf{S} \in \mathbb{R}^{n \times s}$ be the matrix performing this sampling. $\mathbf{S}$ has a single randomly positioned 1 in each column. Thus $\mathbf{KS}$ is an $\mathbb{R}^{n \times s}$ submatrix of $\mathbf{K}$ consisting of randomly sampled columns corresponding to the similarities between all n datapoints and the s landmark points. The key idea is to approximate all pairwise similarities using just this sampled set. In particular, the Nyström approximation of $\mathbf{K}$ is given as:

$$\tilde{\mathbf{K}} = \mathbf{KS}(\mathbf{S}^T\mathbf{KS})^{-1}\mathbf{S}^T\mathbf{K}. \tag{1}$$

Running Time. Observe that the Nyström approximation of (1) requires just $O(ns)$ evaluations of the similarity function to compute $\mathbf{KS} \in \mathbb{R}^{n \times s}$. We typically do not form $\tilde{\mathbf{K}}$ directly, as it would take at least n^2 time to even write down. Instead, we store this matrix in 'factored form', computing $\mathbf{Z} = \mathbf{KS}(\mathbf{S}^T\mathbf{KS})^{-1/2}$. In this way, we have $\mathbf{ZZ}^T = \tilde{\mathbf{K}}$. I.e., the approximate similarity between points x_i and x_j is simply the inner product between the i^{th} and j^{th} rows of $\mathbf{Z}$, which can be thought of as embeddings of the points into $\mathbb{R}^s$. Computing $\mathbf{Z}$ requires computing $(\mathbf{S}^T\mathbf{KS})^{-1/2}$ – the matrix squareroot of $(\mathbf{S}^T\mathbf{KS})^{-1}$ which takes $O(s^3)$ time using e.g., Cholesky decomposition[2]. Multiplying by $\mathbf{KS}$ then takes $O(ns^2)$ time, which is the dominant cost since $n > s$.

Intuition. In (1), $\mathbf{S}^T\mathbf{KS} \in \mathbb{R}^{s \times s}$ is the principal submatrix of $\mathbf{K}$ containing the similarities between the landmark points themselves. To gain some intuition behind the approximation, consider removing the $(\mathbf{S}^T\mathbf{KS})^{-1}$ term and approximating $\mathbf{K}$ with $\mathbf{KSS}^T\mathbf{K}$. That is, we approximate the similarity between any two points x_i and x_j by the inner product between their corresponding rows in $\mathbf{KS}$ – i.e. the vector in $\mathbb{R}^s$ containing their similarities with the landmarks. This would be a reasonable approach – when x_i and x_j are more similar, we expect these rows to have higher dot products.

The $(\mathbf{S}^T\mathbf{KS})^{-1}$ term intuitively 'corrects for' similarities between the landmark points. Formally, when $\mathbf{K}$ is PSD, it can be written as $\mathbf{K} = \mathbf{BB}^T$ for some matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$. Thus $\mathbf{K}_{ij} = \langle \mathbf{b}_i, \mathbf{b}_j \rangle$. Equation (1) is equivalent to projecting all rows of $\mathbf{B}$ onto the subspace spanned by the rows corresponding to the landmark points to produce $\tilde{\mathbf{B}}$, and then letting $\tilde{\mathbf{K}} = \tilde{\mathbf{B}}\tilde{\mathbf{B}}^T$. If e.g., $\text{rank}(\mathbf{K}) \leq s$, then $\text{rank}(\mathbf{B}) = \text{rank}(\mathbf{K}) \leq s$ and so as long as the rows of $\mathbf{B}$ corresponding to the landmark points are linearly independent, we will have $\tilde{\mathbf{B}} = \mathbf{B}$ and thus $\tilde{\mathbf{K}} = \mathbf{K}$. If $\mathbf{K}$ is close to low-rank, as is often the case in practice, $\tilde{\mathbf{K}}$ will still generally yield a very good approximation.

[2] If $\mathbf{S}^T\mathbf{KS}$ is singular, the pseudoinverse $(\mathbf{S}^T\mathbf{KS})^+$ can be used.

2.2 Nyström for Indefinite Matrices

Our extension of the Nyström method to indefinite matrices is motivated by two observations.

Obs. 1: Text Similarity Matrices are Often Close to PSD. Without some form of structure, we cannot approximate a general $n \times n$ matrix in less than $O(n^2)$ time. Fortunately, while many similarity functions used in natural language processing do not lead to matrices with PSD structure, they do lead to matrices that are close to PSD, in that they have relatively few negative eigenvalues, and very few negative eigenvalues of large magnitude. See Figure 1.

Obs. 2: Classic Nyström Fails on Near-PSD Matrices. Given Observation 1, it is natural to hope that perhaps the Nyström method is directly useful in approximating many indefinite similarity matrices arising in NLP applications. Unfortunately, this is not the case – the classic Nyström method becomes very unstable and leads to large approximation errors when applied to indefinite matrices, unless they are very close to PSD. See Figure 3.

A major reason for this instability seems to be that $\mathbf{S}^T\mathbf{KS}$ tends to be ill-conditioned, with several very small eigenvalues that are 'blown up' in $(\mathbf{S}^T\mathbf{KS})^{-1}$ and lead to significant approximation error. See Figure 2. Several error bounds for the classic Nyström method and the related pseudo-skeleton approximation method (where the sampled sets of rows and columns may be different) applied to indefinite matrices depend on $\lambda_{min}(\mathbf{S}^T\mathbf{KS})^{-1}$, and thus grow large when $\mathbf{S}^T\mathbf{KS}$ has eigenvalues near zero (Cai, Nagy, and Xi 2021; Goreinov, Tyrtyshnikov, and Zamarashkin 1997; Kishore Kumar and Schneider 2017). When $\mathbf{K}$ is PSD, by the Cauchy interlacing theorem, $\mathbf{S}^T\mathbf{KS}$ is at least as well conditioned as $\mathbf{K}$. However, this is not the case when $\mathbf{K}$ is indefinite. When $\mathbf{K}$ is indefinite, there may exist well-conditioned principal submatrices. Indeed, a number of methods attempt to select $\mathbf{S}$ such that $\mathbf{S}^T\mathbf{KS}$ is well conditioned (Cai, Nagy, and Xi 2021). However, it is not clear how this can be done in sublinear time in general, without further assumptions.

2.3 Submatrix-Shifted Nyström

Given the above observations, our goal is to give an extension of the Nyström method that can be applied to near-PSD matrices. Our approach is based on a simple idea: if we let $\lambda_{\min}(\mathbf{K})$ denote the minimum eigenvalue of $\mathbf{K}$, then $\bar{\mathbf{K}} = \mathbf{K} - \lambda_{\min}(\mathbf{K}) \cdot \mathbf{I}_{n \times n}$ is PSD. $\bar{\mathbf{K}}$ can thus be approximated with classic Nyström, and if $|\lambda_{\min}(\mathbf{K})|$ is not too large, this should yield a good approximation to $\mathbf{K}$ itself.

There are two issues with the above approach however: (1) $\lambda_{\min}(\mathbf{K})$ cannot be computed without fully forming $\mathbf{K}$ and (2) when $\lambda_{\min}(\mathbf{K})$ is relatively large in magnitude, the shift can have a significant negative impact on the approximation quality – this often occurs in practice – see Figure 1.

We resolve these issues by instead sampling a small principal submatrix of $\mathbf{K}$, computing its minimum eigenvalue, and using this value to shift $\mathbf{K}$. Specifically, consider the Nyström approximation $\mathbf{KS}_1(\mathbf{S}_1^T\mathbf{KS}_1)^{-1}\mathbf{KS}_1$ generated by sampling a set of s_1 indices $S_1 \subseteq [n]$. We let S_2 be a superset of S_1, with size s_2. We typically simply set $s_2 = 2 \cdot s_1$. We

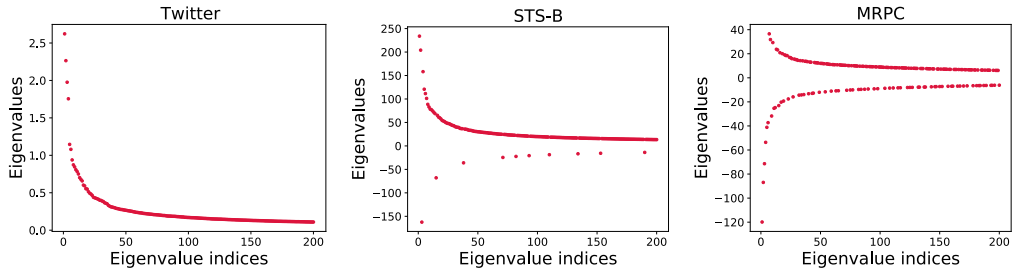

Figure 1: **Eigenspectrums of language similarity matrices**. The eigenspectrums of many text similarity matrices have relatively few negative eigenvalues – i.e., they are relatively close to PSD. Left: similarity matrix arising from the exponentiation of Word Mover's Distance (Kusner et al. 2015) – see Sec. 4.1. Middle and Right: symmetrized cross-encoder BERT sentence and document similarity matrices (Devlin et al. 2018). Eigenvalues are plotted in decreasing order of magnitude from rank 2 to 201. The magnitude of the top eigenvalue is typically very large, and so excluded for better visualization.

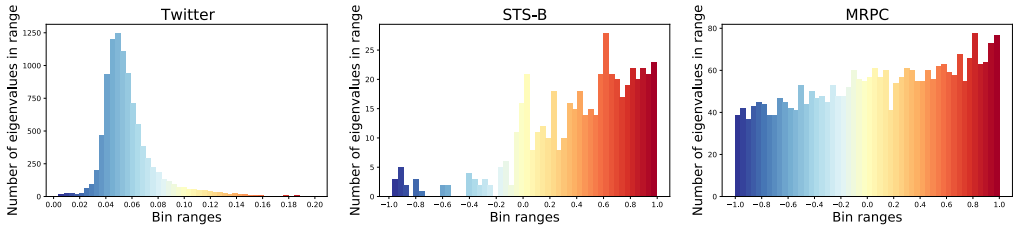

Figure 2: **Eigenvalue histogram plots**. To understand why Nyström fails in indefinite matrices, even when they are relatively near-PSD, we independently sample $\mathbf{S}^T\mathbf{K}\mathbf{S}$ with sample size of 200, 50 times. For each sample we compute all eigenvalues, combine, and plot them in a histogram. As we can see, for the STS-B and MRPC matrices, $\mathbf{S}^T\mathbf{K}\mathbf{S}$ often has eigenvalues very close to zero. For Twitter, which is very near-PSD, there are many fewer eigenvalues very close to zero. As we can see in Figure 3, classic Nyström performs well on Twitter, but fails on the other two matrices.

then compute $e = \lambda_{\min}(\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2)$ and apply the Nyström method to $\bar{\mathbf{K}} = \mathbf{K} - e \cdot \mathbf{I}_{n\times n}$.

Since $\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2$ is a principal submatrix of $\mathbf{K}$, $e = \lambda_{\min}(\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2) \geq \lambda_{\min}(\mathbf{K})$ and thus $\bar{\mathbf{K}}$ will *generally not be PSD*. However, we do have $e \leq \lambda_{\min}(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1)$, since $\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1$ is a submatrix of $\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2$. Thus, $\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1 - e \cdot \mathbf{I}_{n\times n}$ will always be PSD. We also do not expect this matrix to have any very small eigenvalues, since we expect a fairly large gap between $\lambda_{\min}(\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2)$ and $\lambda_{\min}(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1)$ when s_2 is significantly larger than s_1 – e.g. $s_2 = 2 \cdot s_1$. To further insure this, we can multiply e by a small constant factor $\alpha > 1$ (we typically use $\alpha = 1.5$) before applying the shift.

Since $(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1 - e \cdot \mathbf{I}_{n\times n})^{-1}$ is exactly the joining matrix in the Nyström approximation of $\bar{\mathbf{K}}$, our method resolves the issue of small eigenvalues discussed in Sec. 2.2. As we observe in Sec. 3, it is enough to recover the strong performance of Nyström on many near-PSD matrices. Since the minimum eigenvalue of $\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2$ is typically much smaller in magnitude than $\lambda_{\min}(\mathbf{K})$, we often see improved accuracy over the exact correction baseline as well.

We call our method Submatrix-shifted Nyström (SMS-Nyström) and give full pseudocode in Algorithm 1. SMS-Nyström requires roughly the same number of similarity computations and running time as classsic Nyström. We need to perform $(s_2 - s_1)^2$ additional similarity computations to form $\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2$ and must also compute $\lambda_{\min}(\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2)$, which takes $O(s_2^3)$ using a full eigendecomposition. However, this value can also be very efficiently approximated using iterative methods, and typically this additional computation is negligible compared to the full Nyström running time.

Algorithm 1: Submatrix-Shifted Nyström (SMS-Nyström)

Input: Data $\{x_i\}_{i=1}^n \in \mathcal{X}$, sample sizes s_1, s_2, with $s_2 \geq s_1$ scaling parameter α, similarity function $\Delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.
Draw at set of s_2 indices S_2 uniformly at random without replacement from $1, \ldots, n$.
Draw at set of s_1 indices S_1 uniformly at random without replacement from S_2.
$\mathbf{K}\mathbf{S}_1 = \Delta(\mathcal{X}, \mathcal{X}_{S_1})$, $\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1 = \Delta(\mathcal{X}_{S_1}, \mathcal{X}_{S_1})$.
$\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2 = \Delta(\mathcal{X}_{S_2}, \mathcal{X}_{S_2})$.
$e = -\alpha \cdot \lambda_{\min}(\mathbf{S}_2^T\mathbf{K}\mathbf{S}_2)$.
$\mathbf{K}\mathbf{S}_1 = \mathbf{K}\mathbf{S}_1 + e * \mathbf{I}_{n,s_1}$, where $\mathbf{I}_{n\times s_1} \in \mathbb{R}^{n\times s_1}$ has $\mathbf{I}_{ij} = 1$ if $i = j$, $\mathbf{I}_{ij} = 0$ otherwise.
$\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1 = \mathbf{S}_1^T\mathbf{K}\mathbf{S}_1 + e * \cdot\mathbf{I}_{s_1\times s_1}$.
Return $\mathbf{Z} = \mathbf{K}\mathbf{S}_1(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1)^{-1/2}$ with $\mathbf{Z}\mathbf{Z}^T \approx \mathbf{K}$.

3 Matrix Approximation Results

We now evaluate SMS-Nyström and several baselines in approximating a representative subset of matrices.

CUR Decomposition. In addition to the classic Nyström method, we consider a closely related family of *CUR decomposition* methods (Mahoney and Drineas 2009; Wang, Zhang, and Zhang 2016; Pan et al. 2019). In CUR decomposition, the matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ is approximated as the product of a small

subset of columns $\mathbf{KS}_1 \in \mathbb{R}^{n \times s_1}$, a small subset of rows $\mathbf{S}_2^T\mathbf{K} \in \mathbb{R}^{s_2 \times n}$, and a joining matrix $\mathbf{U} \in \mathbb{R}^{s_1 \times s_2}$. $\mathbf{KS}_1$ and $\mathbf{S}_2^T\mathbf{K}$ are generally sampled randomly – the strongest theoretical bounds require sampling according to row/column norms or matrix leverage scores (Drineas, Kannan, and Mahoney 2006; Drineas, Mahoney, and Muthukrishnan 2008). However, these sampling probabilities require $\Omega(n^2)$ time to compute and thus we focus on the setting where the subsets of columns and rows are selected uniformly at random.

There are multiple possible options for the joining matrix $\mathbf{U}$. Most simply and analogously to the Nyström method, we can set $\mathbf{U} = (\mathbf{S}_2^T\mathbf{KS}_1)^+$ – this is also called *skeleton approximation* (Goreinov, Tyrtyshnikov, and Zamarashkin 1997). In fact, if $\mathbf{S}_1 = \mathbf{S}_2$, and $\mathbf{K}$ is symmetric this method is identical to Nyström. Alternatively, as suggested e.g., in (Drineas, Kannan, and Mahoney 2006), we can set $s_1 = s_2 = s$ and $\mathbf{U} = \frac{n}{s} \cdot (\mathbf{KS}_1\mathbf{S}_1^T\mathbf{K})^{-1}\mathbf{S}_1^T\mathbf{KS}_2$. As we will see, these different choices yield very different performance.

Results. We report matrix approximation error vs. sample size for several CUR variants, along with Nyström and SMS-Nyström on the text similarity matrices from Fig. 1, along with a random PSD matrix. Our results are shown in Fig. 3.

- **Nyström.** As discussed in Sec. 2, while Nyström performs well on the PSD matrix and the Twitter matrix, which is very near PSD, it completely fails on the other matrices.
- **SMS-Nyström.** Our simple Nyström variant with $s_2 = 2 \cdot s_1$ and $\alpha = 1.5$ performs well on all test cases, matching the strong performance of Nyström on the PSD and very near-PSD Twitter matrix, but still performing well on the less-near PSD cases of STS-B and MRPC.
- **Skeleton Approximation.** Similar results to Nyström are observed for the closely related skeleton approximation method when $\mathbf{U} = (\mathbf{S}_2^T\mathbf{KS}_1)^+$, $s_1 = s_2$, and $\mathbf{S}_1, \mathbf{S}_2$ are sampled independently. This is unsurprising – this method is quite similar to Nyström.
- **SiCUR.** If we modify the skeleton approximation, using $s_2 > s_1$, we also obtain strong results. Many theoretical bounds for CUR with joining matrix $\mathbf{U} = (\mathbf{S}_2^T\mathbf{KS}_1)^+$ require $s_2 > s_1$ (cf. (Drineas, Mahoney, and Muthukrishnan 2008)), and this choice has a significant effect. It is similar to how SMS-Nyström regularizes the inner matrix – $\mathbf{S}_2^T\mathbf{KS}_1$ is a rectangular matrix whose minimum singular value is unlikely to be too small. We find that setting $s_2 = 2 \cdot s_1$ yields good performance in all cases. To minimize similarity computations, we have $\mathbf{S}_1$ sample a random subset of the indices in $\mathbf{S}_2$. There is very little performance difference if $\mathbf{S}_1$ and $\mathbf{S}_2$ are chosen entirely independently. We call this approach SiCUR for 'Simple CUR'.
- **StaCUR**: Using the $\mathbf{U} = \frac{n}{s} \cdot (\mathbf{KS}_1\mathbf{S}_1^T\mathbf{K})^{-1}\mathbf{S}_1^T\mathbf{KS}_2$ variant of CUR with $s = s_1 = s_2$ yields what we call StaCUR for 'Stable CUR'. StaCUR gives good results on all datasets, however is outperformed by Nyström on PSD matrices and by SMS-Nyström and SiCUR in most other cases. Unlike SMS-Nyström and SiCUR however, StaCUR has no parameters to tune. Unlike for skeleton approximation, setting $s_2 > s_1$ for this method seems to have little effect so we keep $s_1 = s_2$. In Figure 3 we report results for two variants StaCUR(s) and StaCUR(d), where $\mathbf{S}_1, \mathbf{S}_2$ are set equal or to independent samples respectively. StaCUR(s) typically performs better and requires roughly half as many similarity computations, so we use this variant for the remainder of our evaluations.

4 Empirical Evaluation

We now evaluate SMS-Nyström, along with SiCUR and StaCUR on approximating similarity matrices used in document classification, sentence similarity, and cross document coreference, focusing the downstream performance when using the approximated similarity matrix. In each application, we show that our approximation techniques can achieve downstream task performance that matches or is competitive with exact methods, using a fraction of the computation.

4.1 Document Classification with WMD

Our first application is approximating Word mover's distance (WMD) (Kusner et al. 2015) in document classification. WMD is a variant on the Earthmover's distance, which measures how well words in two documents align, based on how far apart they are in a word embedding space. Computing the WMD between two documents with max length L requires $O(L^3 \log(L))$ time (Kusner et al. 2015), and hence computing a full pairwise distance matrix can be very expensive.

Word Movers Embedding. Wu et al. (2018) suggests a PSD similarity function derived from WMD, for which the similarity matrix $\mathbf{K}$ can be approximated very efficiently as $\mathbf{K} \approx \mathbf{ZZ}^T$ using a random features approximation. The resultant feature embeddings $\mathbf{Z}$ are called Word mover's embeddings (WME). Experiments show that WME outperforms true WMD in several classification tasks (Wu et al. 2018).

Our Approach. Following (Wu et al. 2018), we define a similarity function between two documents x, ω by $\Delta(x, \omega) = \exp(-\gamma \text{WMD}(x, \omega))$ for a scalar parameter γ. While this function does not seem to be PSD, it tends to produce near-PSD matrices – see. e.g. the Twitter matrix in Fig. 1. We then approximate the similarity matrix $\mathbf{K}$ using our Nyström and CUR variants. For Nyström, we write $\tilde{\mathbf{K}} = \mathbf{ZZ}^T$ and use $\mathbf{Z}$ as document embeddings (see Alg. 1). For CUR, we factor $\mathbf{U}$ using its SVD $\mathbf{U} = \mathbf{WSV}^T$ as $(\mathbf{WS}^{1/2})(\mathbf{S}^{1/2}\mathbf{V}^T)$, and use $\mathbf{CWS}^{1/2}$ as document embeddings.

Evaluation. We evaluate the performance of our embeddings in multi-class classification for four different corpora drawn from (Huang et al. 2016; Kusner et al. 2015) – Twitter (2176 train, 932 test), Recipe-L (27841 train, 11933 test), Ohsumed (3999 train, 5153 test), and 20News (11293 train, 7528 test). For dataset and hyperparameter details, see App A. We evaluate performance over 20 runs of the respective approximation algorithms for the test set, and for each run we compute the average prediction accuracy and standard deviation.

Following (Wu et al. 2018) we compare the performance of the embeddings produced by WME, SMS-Nyström, SiCUR, and StaCUR at several dimensions (sample sizes s). 'Small Rank', is the dimension ≤ 550 for which the method achieves highest performance. 'Large Rank' is the dimension ≤ 4096 (1500, and 2500 resp. for Twitter and Ohsumed) where the method achieves highest performance. See Table 5 in App. A for the exact values of these ranks. For all except WME,

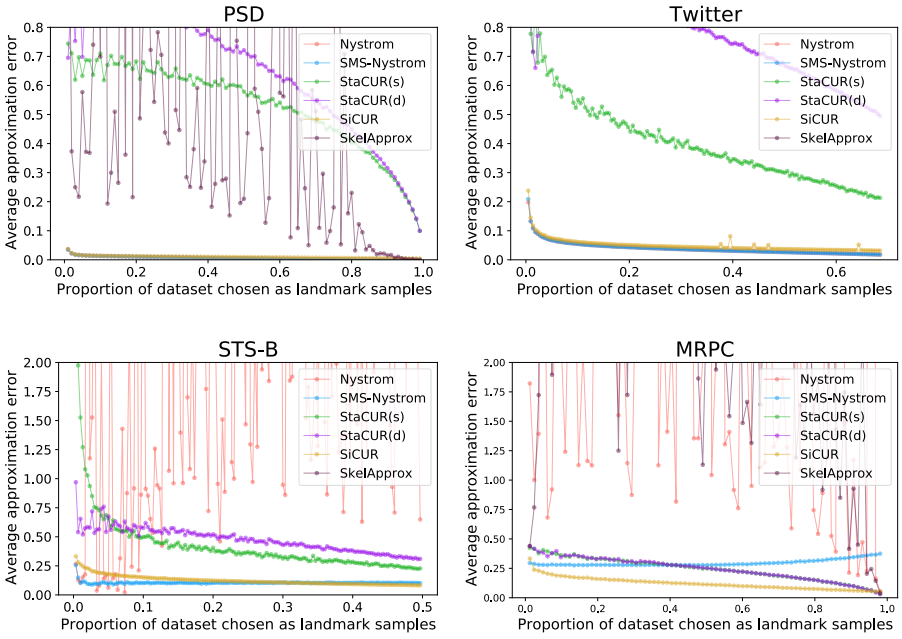

Figure 3: **Approximation error plots.** Evaluation of sublinear time Nyström and CUR variants on the language similarity matrices described in Fig. 1, and a test PSD matrix, $\mathbf{Z}\mathbf{Z}^T$ with $\mathbf{Z} \in \mathbb{R}^{1000\times1000}$ having i.i.d. $\mathcal{N}(0,1)$ entries. Error is reported as $\|\mathbf{K}-\tilde{\mathbf{K}}\|_F/\|\mathbf{K}\|_F$ and averaged over 10 trials. The x-axis is s/n. For SiCUR, where $s_2 > s_1$, it is s_2/n. If a method does not appear, it may be that it had very large error, which is out of range. The error might increase with samples after a certain limit, we believe this is because the correction term overwhelms the approximation error. Zoomed in plots are in Appendix E.

the optimal ranks are typically around the dimension limits. This is expected since the methods achieve higher accuracy in similarity approximation with higher samples.

As baselines, we also compare against (1) WMD-kernel, which uses the true similarity matrix with entries given by $\Delta(x,\omega) = \exp(-\gamma\text{WMD}(x,\omega))$ and (2) Optimal – which uses the optimal rank-k approximation to $\mathbf{K}$ computed with SVD. This method is inefficient, but can be thought of as giving a cap on the performance of our sublinear time methods.

Results. Our results are reported in Table 1. SMS-Nyström consistently outperforms all other methods, and even at relatively low-rank nears the 'optimal' accuracy. In general, the similarity matrix approximation methods tend to outperform the WME baseline. Interestingly, while StaCUR tends to have lower approximation quality on these similarity matrices (see Fig. 3), its performance in downstream classification is comparable to SMS-Nyström and SiCUR.

Observe that the approximation methods achieve much higher accuracy than previous work, WME, including an 8 point improvement on 20News. Our approximation methods achieve results that are within 2-4 points of accuracy of the expensive WMD-kernel true similarity matrix, while maintaining sublinear time and massive space reduction, (especially on corpora like Recipe-L which has tens of thousands of documents). We also observe that SMS-Nystrom and SiCUR can achieve high accuracy for small ranks, compared to both WME and WMD-kernel. The amount of computation we save is considerable, e.g., we require just 14% of the computation for Recipe-L as compared to WMD-kernel. For detailed comparison of rank to performance see App. A.

4.2 Approximation of Cross-Encoder BERT Similarity Matrices

Our second application is to approximate similarity given by a cross-encoder BERT model (Devlin et al. 2018).

Evaluation. We consider three GLUE benchmark datasets –

	Method	Twitter	RecipeL	Ohsumed	20News
Small Rank	WME	72.5 ± 0.5	72.5 ± 0.4	55.8 ± 0.3	72.9
	SMS-N	$\mathbf{75.3 \pm 1.3}$	$\mathbf{77.7 \pm 1.3}$	$\mathbf{59.4 \pm 1.5}$	$\mathbf{79.3 \pm 1.3}$
	StaCUR	73.8 ± 1.5	74.9 ± 1.0	58.7 ± 2.6	76.8 ± 1.6
	SiCUR	74.9 ± 1.5	75.9 ± 1.5	59.3 ± 1.9	73.0 ± 0.6
	Optimal	75.8	78.8	60.3	82.2
Large Rank	WME	74.5 ± 0.5	79.2 ± 0.3	64.5 ± 0.2	78.3
	SMS-N	$\mathbf{76.1 \pm 1.2}$	$\mathbf{80.7 \pm 1.1}$	$\mathbf{65.3 \pm 1.1}$	$\mathbf{86.6 \pm 1.5}$
	StaCUR	71.9 ± 2.3	77.1 ± 1.0	55.7 ± 0.4	84.2 ± 2.1
	SiCUR	75.3 ± 2.1	79.5 ± 1.7	63.3 ± 2.9	85.8 ± 1.0
	Optimal	76.9	81.3	68.2	88.3
WMD-kernel		78.21	82.17	69.03	89.37

Table 1: Results on document classification task with WMD-based similarity. SMS-Nyström is abbreviated as SMS-N.

STS-B, where the goal is to detect sentence similarity, MRPC, where the goal is to detect semantic equivalence, and RTE, where the goal is to detect entailment. See Table 6 in Appendix B for further details. For each task, we first train the BERT model on the test set, using code from (Wolf et al. 2019). We then compute the full BERT similarity matrix for all sentences in the validation set, which consists of a set of sentence pairs, each with a 'true' score, derived from human judgements. The similarity matrices for the datasets STS-B, MRPC and RTE are 3000×3000, 816×816, and 554×554 respectively, and thus are very expensive to fully compute, motivating the use of our fast approximation methods. We compute approximations to this full similarity matrix using SMS-Nyström, SiCUR, and StaCUR. In general, the BERT similarity matrices are non-PSD (see Figure 1), and in fact non-symmetric. So that SMS-Nyström can be applied, we symmetrize them as $\bar{\Delta}(x,\omega) = 1/2 \cdot (\Delta(x,\omega) + \Delta(\omega,x))$.

We use the approximate similarity matrix to make predictions on a dataset of labeled sentences for evaluation. Performance is measured via Pearson and Spearman correlation

	Method	STS-B(P)	STS-B(S)	MRPC	RTE
SMS-Nys	@Rank1	$\mathbf{75.61 \pm 1.3}$@250	$\mathbf{75.27 \pm 1.5}$@250	57.37 ± 2.2@100	60.01 ± 1.1@100
	@Rank2	$\mathbf{77.32 \pm 1.8}$@350	$\mathbf{76.91 \pm 1.8}$@350	63.93 ± 2.7@250	61.84 ± 2.1@250
	@Rank3	$\mathbf{79.36 \pm 1.5}$@700	$\mathbf{78.56 \pm 1.3}$@700	63.04 ± 1.1@500	60.23 ± 1.1@450
StaCUR	@Rank1	28.21 ± 2.3@250	46.77 ± 2.1@250	53.78 ± 4.2@100	58.23 ± 2.2@100
	@Rank2	34.18 ± 1.6@350	49.86 ± 3.2@350	64.41 ± 0.5@250	57.32 ± 1.2@250
	@Rank3	45.87 ± 1.1@700	51.73 ± 1.4@700	66.97 ± 1.1@500	61.37 ± 0.1@450
SiCUR	@Rank1	45.60 ± 3.1@250	44.91 ± 2.8@250	$\mathbf{69.42 \pm 3.7}$@100	$\mathbf{61.11 \pm 2.2}$@100
	@Rank2	57.65 ± 2.6@350	56.52 ± 2.4@350	$\mathbf{72.38 \pm 2.1}$@250	$\mathbf{62.67 \pm 1.5}$@250
	@Rank3	68.84 ± 0.2@700	68.97 ± 0.4@700	$\mathbf{75.53 \pm 0.9}$@500	$\mathbf{63.28 \pm 0.3}$@450
	BERT	85.09	84.70	83.30	65.98
	SYM-BERT	85.54	85.13	83.75	66.10

Table 2: Performance comparison of original BERT similarities and approximated similarities on GLUE benchmarks. Ranks (i.e., sample size) are recorded next to each result.

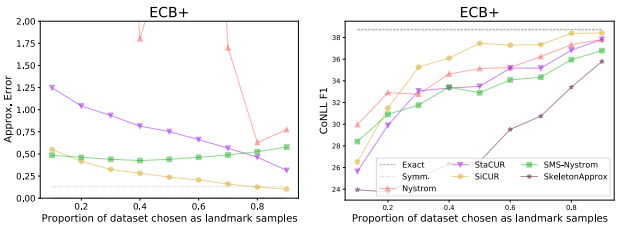

Figure 4: **Cross-document Entity & Event Coreference Performance**. We report the downstream task F1 performance and approximation error on EventCorefBank (ECB+).

with the human scores for STS-B, F1 score of predicted labels for MRPC, and accuracy for RTE. We report the average scores obtained with different sample sizes, over 50 runs.

Results. Table 2 reports results for the approximations, the exact, and the symmetrized (SYM-BERT) approaches. SMS-Nyström performs particularly well on STS-B, while SiCUR performs best on MRPC. All methods are comparable on RTE. This performance is inline with the accuracy in approximating $\mathbf{K}$, which is reported in Appendix B.

4.3 Approximate Similarity Matrices for Entity & Event Coreference

Cross-document entity and event coreference is a clustering problem. Ambiguous mentions of entities and events that appear throughout a corpus of documents are to be clustered into groups such that each group refers to the same real world entity or event. Cattan et al. (2020) present an approach that (1) learns a pairwise similarity function between ambiguous mentions and (2) uses average-linkage agglomerative clustering with a similarity threshold to produce the predicted clustering. The pairwise similarity function is a MLP which takes as input the concatenation of RoBERTa (Liu et al. 2019), embeddings of two mentions and their elementwise product. This induces a matrix that is asymmetric and not-PSD. We symmetrize the matrix for the approximations.

Evaluation. We evaluate both the approximation error as well as the downstream coreference task performance (CoNLL F1 (Pradhan et al. 2014)) of approximating similarity matrix of the model. We evaluate on the EventCorefBank+ Corpus (Cybulska and Vossen 2014) See App. C for details.

Results. Figure 4 shows the downstream task performance measured in CoNLL F1 and the approximation error as a function of the number of landmarks used. We find a similar trend as the previous two tasks. SiCUR performs very well in terms of both metrics, with performance improving as more landmarks are added, achieving nearly the same F1 (within 1 point) performance when 90% of the data is used for landmarks and very competitive performance (within 1.5 points) with just 50%, a drastic reduction in time/space compared to the exact matrix. SMS-Nyström required additional rescaling for this task likely due to sensitivity of threshold of agglomerative clustering. We report the rescaled version, which is quite competitive with StaCUR (see Appendix C for more detail). The results indicate that the proposed approximation could help scale models for which the $\Omega(n^2)$ similarity computations would be intractable.

5 Conclusion

We have shown that indefinite similarity matrices arising in NLP applications can be effectively approximated in sublinear time. A simple variant of the Nyström method, and several simple CUR approximation methods, all display strong performance in a variety of tasks. We hope that in future work, these methods can be used to scale text classification and clustering based on cross-encoder, word mover's distance, and other expensive similarity functions, to much larger corpora.

Acknowledgements

This work supported in part by the Center for Data Science, in part the Center for Intelligent Information Retrieval, in part by the National Science Foundation under Grants No. 1763618. Some of the work reported here was performed using high performance computing equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Bakshi, A.; and Woodruff, D. P. 2018. Sublinear Time Low-Rank Approximation of Distance Matrices. *Advances in Neural Information Processing Systems 31 (NeurIPS)*.

Belongie, S.; Fowlkes, C.; Chung, F.; and Malik, J. 2002. Spectral partitioning with indefinite kernels using the Nyström extension. In *European Conference on Computer Vision*.

Bentivogli, L.; Clark, P.; Dagan, I.; and Giampiccolo, D. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.

Cai, D.; Nagy, J.; and Xi, Y. 2021. Fast and stable deterministic approximation of general symmetric kernel matrices in high dimensions. *arXiv:2102.05215*.

Cattan, A.; Eirew, A.; Stanovsky, G.; Joshi, M.; and Dagan, I. 2020. Streamlining Cross-Document Coreference Resolution: Evaluation and Modeling. *arXiv:2009.11032*.

Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv:1708.00055*.

Chen, Y.; Gupta, M. R.; and Recht, B. 2009. Learning kernels from indefinite similarities. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26 (NeurIPS)*.

Cybulska, A.; and Vossen, P. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In *LREC*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Dolan, W. B.; and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Dong, W.; Moses, C.; and Li, K. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th International World Wide Web Conference (WWW)*.

Drineas, P.; Kannan, R.; and Mahoney, M. W. 2006. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*.

Drineas, P.; Mahoney, M. W.; and Cristianini, N. 2005. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *Journal of Machine Learning Research*.

Drineas, P.; Mahoney, M. W.; and Muthukrishnan, S. 2008. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*.

Frieze, A.; Kannan, R.; and Vempala, S. 2004. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*.

Gionis, A.; Indyk, P.; Motwani, R.; et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*.

Gisbrecht, A.; and Schleif, F.-M. 2015. Metric and non-metric proximity transformations at linear costs. *Neurocomputing*.

Gittens, A.; and Mahoney, M. W. 2016. Revisiting the Nyström method for improved large-scale machine learning. *The Journal of Machine Learning Research*.

Goreinov, S. A.; Tyrtyshnikov, E. E.; and Zamarashkin, N. L. 1997. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*.

Huang, G.; Guo, C.; Kusner, M. J.; Sun, Y.; Sha, F.; and Weinberger, K. Q. 2016. Supervised Word Mover's Distance. *Advances in Neural Information Processing Systems 29 (NeurIPS)*.

Huang, P.-S.; Avron, H.; Sainath, T. N.; Sindhwani, V.; and Ramabhadran, B. 2014. Kernel methods match deep neural networks on TIMIT. In *Proceedings of the 2014 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Humeau, S.; Shuster, K.; Lachaux, M.-A.; and Weston, J. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv:1905.01969*.

Hwang, Y.; Han, B.; and Ahn, H.-K. 2012. A fast nearest neighbor search algorithm by nonlinear embedding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Indyk, P.; Vakilian, A.; Wagner, T.; and Woodruff, D. P. 2019. Sample-optimal low-rank approximation of distance matrices. In *Proceedings of the 32nd Annual Conference on Computational Learning Theory (COLT)*.

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2019. TinyBERT: Distilling BERT for natural language understanding. *arXiv:1909.10351*.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kishore Kumar, N.; and Schneider, J. 2017. Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra*.

Kumar, S.; Mohri, M.; and Talwalkar, A. 2012. Sampling methods for the Nyström method. *The Journal of Machine Learning Research*.

Kusner, M.; Sun, Y.; Kolkin, N.; and Weinberger, K. 2015. From word embeddings to document distances. In *International conference on machine learning*.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv:1909.11942*.

Le, Q.; Sarlós, T.; and Smola, A. 2013. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Lv, Q.; Josephson, W.; Wang, Z.; Charikar, M.; and Li, K. 2007. Multi-probe LSH: Efficient indexing for high-dimensional similarity search. In *VLDB*.

Mahoney, M. W. 2011. Randomized algorithms for matrices and data. *arXiv:1104.5557*.

Mahoney, M. W.; and Drineas, P. 2009. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*.

Meanti, G.; Carratino, L.; Rosasco, L.; and Rudi, A. 2020. Kernel methods through the roof: handling billions of points efficiently. *Advances in Neural Information Processing Systems 33 (NeurIPS)*.

Michel, P.; Levy, O.; and Neubig, G. 2019. Are sixteen heads really better than one? *Advances in Neural Information Processing Systems 32 (NeurIPS)*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Musco, C.; and Musco, C. 2017. Recursive sampling for the Nyström method. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*.

Musco, C.; and Woodruff, D. P. 2017. Sublinear time low-rank approximation of positive semidefinite matrices. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*.

Oglic, D.; and Gärtner, T. 2019. Scalable learning in reproducing kernel Krein spaces. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Orchard, M. T. 1991. A fast nearest-neighbor search algorithm. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Pan, V. Y.; Luan, Q.; Svadlenka, J.; and Zhao, L. 2019. CUR Low Rank Approximation of a Matrix at Sublinear Cost. *arXiv:1906.04112*.

Piccoli, B.; and Rossi, F. 2014. Generalized Wasserstein distance and its application to transport equations with source. *Archive for Rational Mechanics and Analysis*.

Pradhan, S.; Luo, X.; Recasens, M.; Hovy, E.; Ng, V.; and Strube, M. 2014. Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Rahimi, A.; and Recht, B. 2007. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems 20 (NeurIPS)*.

Rastogi, P.; Cotterell, R.; and Eisner, J. 2016. Weighting Finite-State Transductions With Neural Context. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*.

Rubner, Y.; Tomasi, C.; and Guibas, L. J. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108*.

Sarlos, T. 2006. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*.

Schleif, F.-M.; Gisbrecht, A.; and Tino, P. 2018. Supervised low rank indefinite kernel approximation using minimum enclosing balls. *Neurocomputing*.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and De Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*.

Talwalkar, A.; and Rostamizadeh, A. 2014. Matrix coherence and the Nyström method. *arXiv:1408.2044*.

Tam, D.; Monath, N.; Kobren, A.; Traylor, A.; Das, R.; and McCallum, A. 2019. Optimal Transport-based Alignment of Learned Character Representations for String Similarity. In *Association for Computational Linguistics*.

Wang, S.; and Zhang, Z. 2013. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*.

Wang, S.; Zhang, Z.; and Zhang, T. 2016. Towards more efficient SPSD matrix approximation and CUR matrix decomposition. *The Journal of Machine Learning Research*.

Williams, C.; and Seeger, M. 2001. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 14 (NeurIPS)*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771*.

Woodruff, D. P.; et al. 2014. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*.

Wu, L.; Yen, I. E.; Xu, K.; Xu, F.; Balakrishnan, A.; Chen, P.-Y.; Ravikumar, P.; and Witbrock, M. J. 2018. Word mover's embedding: From word2vec to document embedding. *arXiv:1811.01713*.

Wu, L.; Yen, I. E.-H.; Zhang, Z.; Xu, K.; Zhao, L.; Peng, X.; Xia, Y.; and Aggarwal, C. 2019. Scalable global alignment graph kernel using random features: From node embedding to graph embedding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Yang, T.; Li, Y.-F.; Mahdavi, M.; Jin, R.; and Zhou, Z.-H. 2012. Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in Neural Information Processing Systems 25 (NeurIPS)*.

Zafrir, O.; Boudoukh, G.; Izsak, P.; and Wasserblat, M. 2019. Q8BERT: Quantized 8bit BERT. *arXiv:1910.06188*.

Zhang, K.; Tsang, I. W.; and Kwok, J. T. 2008. Improved Nyström low-rank approximation and error analysis. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.