An Efficient Framework for Constructing Generalized Locally-Induced Text Metrics

Saeed Amizadeh
Intelligent Systems Program
University of Pittsburgh
saa63@pitt.edu

Shuguang Wang
Intelligent Systems Program
University of Pittsburgh
swang@cs.pitt.edu

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh
milos@pitt.edu

Abstract

In this paper, we propose a new framework for constructing text metrics which can be used to compare and support inferences among terms and sets of terms. Our metric is derived from data-driven kernels on graphs that let us capture global relations among terms and sets of terms, regardless of their complexity and size. To compute the metric efficiently for any two subsets of terms, we develop an approximation technique that relies on the precompiled term-term similarities. To scale-up the approach to problems with huge number of terms, we develop and experiment with a solution that subsamples the term space. We demonstrate the benefits of the whole framework on two text inference tasks: prediction of terms in the article from its abstract and query expansion in information retrieval.

1 Introduction

A huge number of text documents is published or shared every day in different areas of science, technology, culture etc. Due to the huge volumes, the analysis of these documents and their contents is becoming increasingly hard. This prompts the development of tools that let us better analyze these texts and support various automatic inferences upon their content.

This paper focuses on the development of a new class of text kernels that let us capture complex relations between terms and sets of terms in a corpus. Such kernels can be very useful for analysis of term relations, or to support inference tasks such as term predictions, term clustering, or more applied tasks such as query expansion. The key challenge in designing a good text kernel is to account for indirect global term relations that are not immediate from relations explicitly mentioned in the text. As an example, there is a logical relevance between 'designer' and 'software' and between 'designer' and 'cloth', while 'software' and 'cloth' may not ever happen together in a document while there is a weak similarity between them as artifacts designed by humans.

At the core of our methodology is the design of term-term similarity metrics (or, in other words, term-term kernels). These metrics aim to capture abstract and often complex relations among terms and their strength. All the proposed metrics in this paper are derived from a graph of local associations among the terms observed in the document corpus. We call this graph the *association graph*. To cover and account for indirect relations which span multiple direct associations, we define a global term similarity metric based on the spectral decomposition of the association graph and a special class of graph-Laplacian kernels [Zhu *et al.*, 2006] that assure the smoothness of the metric across observed term associations.

The term-term similarity metric is defined on the term space. However, many useful text inferences, e.g. query expansion, work with sets of terms. To address the problem, we need a *generalized* term similarity metric that lets us represent set-set similarities. We show how this new metric can be, in principle, built by expanding the original association graph with n nodes (corresponding to terms) with special auxiliary nodes representing sets of terms. However, computing the distance between any two sets would require a new graph expansion and the recalculation of the $O(n^3)$ spectral decomposition, which is infeasible in practice. We approach the problem by proposing and defining a new method that can efficiently approximate the set-set similarities on-line, whenever they are needed, by computing the spectral decomposition of the underlying graph only once.

The spectral decomposition of the graph Laplacian takes $O(n^3)$ time. This is prohibitive for large n even if computed only once. One way to alleviate the issue is to benefit from the graph structure and its disconnected components. A more principled approach requires a reduction of the number of terms in the graph. We propose and study an approximation approach that first performs the decomposition on a randomly selected sub-graph of the association graph, and then extends the results to the entire graph to approximate the spectral decomposition of the full graph.

As mentioned earlier, a good metric relating terms and their sets can be used to support various text inferences such as text clustering, query expansion, etc. We demonstrate the benefit of our text kernels on two text inference problems: (1) prediction of terms in the full document from terms in its abstract and (2) the query expansion for the retrieval of documents relevant to the search query [G. Cao and Robertson, 2008].

2 Related Work

Embedding-based methods for text metric learning have been recently studied in the literature. In [Lebanon, 2006], parametric metrics have been learned for text documents based

on Fisher geometry. In [Cai *et al.*, 2005], text documents are mapped into a semantic space using Locally Preserving Indexing. However, in both frameworks, the text metrics only compare documents and not any arbitrary chunks of texts.

Laplacian kernels and their special cases, on the other hand, have been used in various machine learning problems. However, their application in text analysis is still relatively rare. [Dillon *et al.*, 2007] used a specific Laplacian kernel to derive the document similarity for Machine Translation tasks. [Bordino *et al.*, 2010] proposed to estimate the query similarity using the classical spectral projection on the query flow graph. This is in fact a special case of Laplacian embedding using step function transform which is used on the graph over queries instead of terms. Our framework is more general than these two works and also assures the metric is smooth in that it respects directly the observed term relations.

[Collins-Thompson and Callan, 2005] proposed to model term association using the Markov chain (random walk) model defined on the graph over terms. The edges in the graph are defined using multiple external sources such as the synonyms in WordNet [Moldovan and Rus, 2001]. The analysis is performed on the term relevance graph directly instead of its Laplacian matrix. We propose to model term relevance using a general family of graph-Laplacian kernels.

We note that there are different ways of defining term similarity other than graph-based methods. For example, [Wang *et al.*, 2010] propose to apply a PHITS model, originally designed for the link analysis of document networks, to term association graphs. The PHITS model learned from the training data is then used to approximate the probability of a term-term association. However, this method projects terms into a latent low-dimensional space representing clusters of interconnected terms and the similarity for any pair is computed in this space. In our experiments, we show that these methods are outperformed by our graph-based similarity metrics.

For computing the similarity between sets of objects, [Kondor and Jebara, 2003] proposed to compute the Bhattacharyya divergence between the densities of the sets in the embedded space. [Bach, 2008] proposed to use efficient graph kernels to compute the kernel between point clouds. Our set similarity extension is different from these methods in that first we work with sets of terms (and not vectors) and second our work is inspired by short-circuiting in electrical circuits.

3 Laplacian-based Graph Kernels

The framework we propose in this paper computes the similarity between two texts using Laplacian-based graph kernels. In this section, we briefly review the basics of the graph Laplacian and data-driven kernels derived from it. Interested readers may refer to [Chung, 1997] for further details.

Let $G = \langle V, E, W \rangle$ be a weighted graph with a weighted adjacency matrix W, such that $W_{ij} = w(e_{ij})$ is the weight of edge $e_{ij} \in E$ and 0 otherwise. If D is the diagonal matrix with diagonal entries $D_{jj} = \sum_k W_{jk}$, the Laplacian L of G is defined as: $L = D - W$ [Ng *et al.*, 2001]. L is a semi-positive definite matrix with the smallest eigenvalue $\lambda_1 = 0$ [Chung, 1997]. The eigen decomposition of L is $L = \sum_i \lambda_i u_i u_i^T$, where λ_i and u_i denote eigenvalues and their respective eigenvectors.

Let $f : V \to \mathbb{R}^k$ be a vector-valued function over the nodes of G. Let the smoothness of $f(\cdot)$ w.r.t. G be defined as:

$$\Delta_G(f) \triangleq f^T L f = \frac{1}{2} \sum_{e_{ij}} W_{ij} \|f(i) - f(j)\|_2^2 \qquad (1)$$

The smoother the function $f(\cdot)$ is on G ($f(i)$ and $f(j)$ are close in $\mathbb{R}^k$ if nodes i and j are close in G), the smaller is $\Delta_G(f)$. By replacing L with its eigen decomposition, we will get $\Delta_G(f) = \sum_i \lambda_i a_i^2$ where $a_i = f^T u_i$ is the projection of $f(\cdot)$ on the i^{th} eigenvector of L. For $f = u_k$, we will have $\Delta_G(f) = \lambda_k$. Therefore, those eigenvectors of L with small eigenvalues, in fact, define smooth functions over G.

To define a similarity metric on V, a Laplacian-based kernel matrix K is defined to be a semi-positive definite kernel with exactly the same eigenvectors as L and different non-negative eigenvalues $\boldsymbol{\theta} = [\theta_i]_{n \times 1}$; that is, $K = \sum_i \theta_i u_i u_i^T$. Depending on the values of $\boldsymbol{\theta}$, K can define very different similarity metrics on V. Here, we are interested in those kernels which make nodes close in G more similar. This is equivalent to assigning more weight (eigenvalue θ) to the smoother eigenvectors of L in building the kernel K. To this end, Zhu et al. [Zhu *et al.*, 2006] define a class of Laplacian-based graph kernels with $\theta_i = g(\lambda_i)$, where $g(\cdot)$ is an arbitrary *spectral transformation* function which is a non-negative non-increasing function of λ_is, the eigenvalues of L. This last condition assures that smoother eigenvectors (with smaller λ) are assigned higher weights in the kernel. In fact, the reason that we are interested in Laplacian-based kernels as opposed to more general kernels is that their eigen decomposition gives us a series of eigenfunctions on the graph sorted by their degrees of smoothness.

Given a kernel K, now we can define the distance between nodes i and j in the graph as $d(i,j) = K_{ii} + K_{jj} - 2K_{ij}$. Furthermore, any Laplacian-based kernel K defines a mapping $\phi : V \to \mathbb{R}^n$ from the nodes of the graph to some metric feature space such that $K_{ij} = \phi(i)^T \phi(j)$, $d(i,j) = \|\phi(i) - \phi(j)\|_2^2$ where:

$$\phi(i) = \left[\sqrt{\theta_1} u_1(i), \sqrt{\theta_2} u_2(i), \ldots, \sqrt{\theta_n} u_n(i)\right]^T \qquad (2)$$

$u_k(i)$ denotes the i^{th} element of the k^{th} eigenvector of L.

A variety of kernels can be defined using different spectral transformation functions $g(\cdot)$. For example, $g(\lambda) = 1/(\lambda + \epsilon)$ for some small $\epsilon > 0$ (called resistance kernel) is the kernel whose derived distance measure approximates the total resistance between two nodes in G, given that the edge weights are interpreted as electric conductances (reciprocal of electric resistances) [Doyle and Snell, 1984; Klein and Randić, 1993]. Other useful kernels in this class are the diffusion kernel: $g(\lambda) = \exp(-\lambda\sigma^2/2)$ [Kondor and Lafferty, 2002] and the random walk kernel: $g(\lambda) = (\alpha - \lambda)$ [Zhu *et al.*, 2006], where σ^2 and $\alpha \geq 2$ are parameters of these kernels. Alternatively, one can define a non-increasing function $g(\cdot)$ in a completely non-parametric fashion by setting the values of θ_is independently without using any functional form (we refer to it as non-parametric kernel). In principle, given a relatively smooth function $f(\cdot)$ on G, one can

optimize the kernel parameters or the values of θ_is directly such that the similar nodes according to $f(\cdot)$ become closer according to K as well.

4 The Graph-based Text Metric

We build our framework for generating text distance metrics based on the Laplacian-based graph kernels. In particular, we first show how a distance metric between the simplest elements of texts, namely terms, can be induced and then generalize it to define a distance metric between the sets of terms.

4.1 Term-Term Distance Metrics

First, let us define the key ingredients of our framework.

Term association graph is a weighted graph $\mathcal{A}$ with nodes V corresponding to the terms in the lexicon extracted from an input document corpus. The edges in $\mathcal{A}$ represent the co-occurrence of the two terms corresponding to the edge in the same sentences. Furthermore, each edge is assigned an *association weight* in $\mathbb{R}^+$ expressing the strength of the relation. For the term co-occurrence relation, the strength is the number of different documents in which the two terms co-occur in the same sentence. We note that, in general, the co-occurrence relation and its weight can be replaced with any reasonable term-term relation and corresponding statistics as long as it is easy to extract them from the input corpus.

Relevance function is defined as a vector-valued function $r : V \mapsto \mathbb{R}^k$ such that if two terms t_i and t_j are considered to be *relevant* according to the domain of the desired task, then $\|r(t_i) - r(t_j)\|_2^2$ is small. Thus, knowing $r(\cdot)$ would greatly help to build a reliable term-term distance metric. However, in general, the true $r(\cdot)$ is unknown for a given problem.

Now, the question is how $r(\cdot)$ and $\mathcal{A}$ are related? Our key assumption in this work is: $r(\cdot)$ *is relatively smooth with respect to* $\mathcal{A}$*; that is, the smoothness* $\Delta_{\mathcal{A}}(r)$ *is relatively small.* As a result, we can use the Laplacian-based kernels derived from $\mathcal{A}$ to define term-term distance metrics which are able to capture the true relevance among the terms. In particular, we use the resistance, diffusion and non-parametric kernels in the previous section to build distance metrics on terms.

As mentioned before, the parameters of these kernels can be optimized based on the task (or equivalently the true $r(\cdot)$). However, since $r(\cdot)$ is unknown, we can use some proxy objective for this optimization. In particular, if we have the binary information whether two terms are relevant or not on a subset of terms as training data, we can maximize the AUC measure between the goal standard and the distance ordering derived from the kernel on the same set of terms. Based on this objective, the optimization of single-parameter kernels, such as the diffusion kernel, can be carried out using a simple line search procedure. For the non-parametric kernel (with spectral transformation parameters θ_i subject to constraints), we define a linear program to find the optimal $\boldsymbol{\theta}$ vector as:

$$\max_{\boldsymbol{\theta}=(\theta_1,\ldots,\theta_n)^T} \quad \sum_{ij} K(t_i,t_j) - b\boldsymbol{\theta}^T\Lambda$$
$$\text{s.t.} \quad 0 \le \theta_i \le \theta_{i+1} \le 2 \; \forall i = n-1,\cdots,1$$

where the sum is over all pairs of terms which are considered relevant according to the training data, $b \ge 0$ is a regularization penalty and Λ is the vector of eigenvalues of the $\mathcal{A}$'s Laplacian. The order constraints over θs assure that smoother eigenvectors are assigned higher weights.

Now that the kernel is specified and its parameters are optimized, one can derive the mapping $\phi(\cdot)$ using Eq. (2) to define the distance between terms. In fact, $\phi(\cdot)$ can be seen as an approximation to the true $r(\cdot)$.

4.2 Set-Set Distance Metric

To generalize the distance measures derived in the previous subsection to the distance between sets of terms, a straightforward approach is to somehow combine the mutual term-term distances between the two sets. To do so, the natural candidates are the *max*, the *min* and the *average* functions. Here, we develop a more principled approach to compute the distance between two sets of terms.

Recall that the resistance kernel in the previous section approximates the total resistance between two nodes in the graph if the edge weights are interpreted as reciprocal of resistance. In an actual circuit, in order to compute the total resistance between two sets of nodes S_1 and S_2, one should first short-circuit all the nodes in each set separately to collapse each set to one node. Then, the total resistance between the collapsed nodes is equal to the total resistance between S_1 and S_2. Figures 1(I)&(II) illustrate the idea. Figure 1(I) shows an electrical network; Figure 1(II) is the same network after collapsing the terms (nodes) in the set $S = \{A, E\}$.

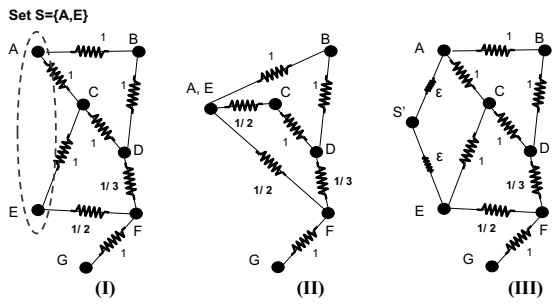

Figure 1: Collapsing nodes in electrical resistance network

Note that the electrical circuit example is just one analogy and the core idea is more general. In fact, short-circuiting the nodes in a set S in an electrical circuit is equivalent to adding high association (0 resistance) edges between S's members in a more general graph. After doing so, if a candidate node x is similar to *any* of S's members then it will become similar to all the nodes in S (due to the insertion of high association edges). This somehow encodes an 'OR' logic.

We extend the same idea to compute the distance between two sets of terms for any Laplacian-based kernel. That is, to compute the distance between the term sets S_1 and S_2, we collapse the corresponding terms of each set in $\mathcal{A}$ to one super node to obtain a new reduced association graph $\mathcal{A}'$. After that we recompute the Laplacian-based kernel for $\mathcal{A}'$ to get the distance between the collapsed nodes. The main drawback of this approach is for any given two subsets of nodes, we have to reshape the graph, form the Laplacian matrix and compute its eigen decomposition which takes $O(n^3)$ time.

To avoid recalculations, we propose an efficient approximation that does not require us to change the structure in the underlying association graph. The solution is based on the following electrical circuit analogy: short-circuiting the nodes in some set S is equivalent to adding an auxiliary node s' to the network and connecting it to the nodes in S with zero resistance (infinite association weight) links. The total resistance between s' and any other node in the graph is equivalent to the resistance between the collapsed node representing S to these nodes. We apply the same trick on the association graph to build our approximation: instead of collapsing the terms in a set into one node, we add a new node s' to $\mathcal{A}$ and connect it to all terms in S with some high association weights (to approximate infinite weights). This is illustrated in Figure 1(III). Now, we have to compute the eigen decomposition for the Laplacian of $\mathcal{A} \cup s'$ to find the mapping $\phi(s')$ for the new node; however, instead of recalculating everything from scratch, we can just extend the eigenvectors of $\mathcal{A}$'s Laplacian to one more element using the Nyström approximation [Buchman *et al.*, 2009]. More specifically, if node s' was included in the graph, we would have $\sum_j L(s', j)u_k(j) = \lambda_k u_k(s')$, for the k^{th} eigenvector of L. Solving for $u_k(s')$, we will get:

$$\forall k, u_k(s') = \frac{1}{\lambda_k - L(s', s')} \sum_{j \neq s'} L(s', j)u_k(j), \quad (3)$$

where $L(s', j)$ is just the negative assigned association weight between node j and the auxiliary node s' (and 0 if $j \notin S$). Also $L(s', s')$ is the degree of s'. Having approximated $u_k(s')$ for all k, we can compute $\phi(s')$ using Eq. 2.

Using this approximation, we propose to define and calculate the set-set kernel as follows. First, we compute and store the eigen decomposition for the term-term associations, which takes $O(n^3)$ time and $O(n^2)$ space. This step is performed offline. The metric for any two sets of terms S_1 and S_2 is then calculated online using the stored decomposition. In fact, it only takes $O(|S_1| + |S_2|)$ to find $\|\phi(s_1) - \phi(s_2)\|_2^2$.

4.3 Scaling Up to Large Association Graphs

If the number of terms n is large, computing the eigen decomposition over the entire term space (in principle $O(n^3)$ time) becomes computationally demanding, even if it is done offline and only once. To address this issue, we propose to first build $\mathcal{A}$ over the whole term space and then keep only a random sub-graph of size m for the eigen decomposition purposes. We can use the same Nyström approximation as used before to extend the eigenvectors of the sub-graph (size m) to vectors of size n over the entire graph. Using this approximation, the total time reduces to $O(m^3 + \bar{d}n)$, where $\bar{d}$ is the average node degree in $\mathcal{A}$. This is a significant improvement if $m \ll n$.

Obviously, eigen-decomposition of a smaller graph and its Nyström-based expansion to all nodes define an approximation of the true metric. Now, we have to show how much this approximation affects the final distance metric. However, in many real applications, it is the ordering over the terms induced by the distance metric that really matters and not the actual distance values. Therefore, we can only measure how many misplacements (in %) are introduced using the proposed approximation compared to the ordering induced in the exact case (i.e. the gold standard order). Table 1 shows the (normalized) number of misplacements introduced in a test set of 100 terms using different sample sizes m for a term space and the graph of size $n = 5000$ we used for the analysis of PubMed articles in Section 5. The experiment is repeated 5 times for each sample size and the results are averaged. As the table shows, even for 20% of the terms ($m = 1000$), 16% misplacements are introduced in average which means that the approximation is pretty robust.

m	Avg % of misplacements	std deviation
20%	16.0%	0.011
40%	13.7%	0.005
60%	8.9%	0.012
80%	4.4%	0.002

Table 1: The average number of misplacements for different sample size m with standard deviation

5 Experiments

A good text similarity metric can open door to some interesting applications: predicting term occurrences from text components, clustering of text components, query expansion, etc. In this section, we demonstrate the merits of our framework on two applications; prediction of terms in the document and query expansion in information retrieval.

5.1 Term Prediction

The objective of this experiment is to demonstrate that our kernel-based distance metrics can predict the occurrence of terms in a full article from terms in the abstract. Intuitively, for the same document, terms in the full body should be very relevant to the terms mentioned in its abstract.

Data: The documents used in this experiments are from the cancer corpus [Wang and Hauskrecht, 2008] that consists of 6,000 documents related to 10 cancer subtypes that are indexed in PubMed. The articles were randomly divided into the training (80%) and test (20%) sets. Only the abstracts in the training set were used to build the term association network. Although, we could have trained our kernels using the terms in the document bodies as well, they perform well over the entire vocabulary even just using the terms in the abstracts. The terms extracted were the names of genes and proteins occurring in the free text. We used LingPipe[1] to identify genes and proteins.

Evaluation Metric: For evaluation, we first compute the distances between terms in the abstracts to all candidate terms and rank them. If our proposed similarity metrics is good, the terms in the full body of the text should be ranked higher than the rest. We assess this using the Area Under the ROC Curve (AUC) score. More specifically, we assign label 0 to those concepts that were not observed in the full article and 1 to those that were observed. The ranking based on the metric

[1] http://alias-i.com/lingpipe

is then combined with the labels and the AUC score is calculated. Note that the optimal term ordering for this document should give a perfect separation of 1s and 0s.

Baselines: We compare our methods to three baseline methods: TF-IDF, PHITs, and the shortest-path approach. The TF-IDF method predicts the terms in the document body using the TF-IDF score ranking [Salton and McGill, 1983] calculated on the training documents and is independent of the query (the terms in the abstract). The PHITS-based approach [Wang and Hauskrecht, 2008] first learns the PHITS model [Cohn and Chang, 2000] from the term association graph and uses it to approximate the strength of the term and set-to-term relations. The shortest path method uses the term association graph and its reciprocal weights to calculate the shortest paths between terms in the abstract and the rest of the terms. The shortest path lengths are then used to estimate term-term and set-to-term similarities.

Results: Table 2 summarizes the results of the experiment on the *full* term vocabulary of 1200 test documents. For each method, the table shows the mean AUC scores obtained for test documents and their 95% confidence intervals (CI).

Methods	AUC	95% CI
TF-IDF	0.782	[0.767, 0.801]
PHITS	0.781	[0.749, 0.805]
shortest-path	0.745	[0.729, 0.756]
$K_{Diffusion}$	0.878	[0.870, 0.887]
$K_{Resistance}$	**0.883**	[0.878, 0.892]
$K_{Nonpara}$	0.870	[0.863, 0.878]

Table 2: AUCs for predicting terms on test documents. The best AUC score is in bold.

Baselines vs. Kernels: All Laplacian-based metrics were statistically significantly better than baselines when predicting the terms in full documents. This means the derived similarity metrics are very meaningful and model the term relevance better than baselines.

Comparison of Kernels: The parameters of all kernels were optimized using either line search (for diffusion and resistance kernels) or the linear program (for non-parametric kernel). There are small overlaps between confidence intervals of different kernel methods. To examine the differences in the mean AUC scores more carefully, we analyzed the methods using pair-wise comparisons. We found that the resistance kernel performs statistically significantly better than other kernels. The diffusion kernel and the non-parametric kernel were not significantly different. We attribute the superiority of the resistance kernel to the fact that it heavily emphasizes the smoother (smaller) eigenvalues of the Laplacian compared to other kernels due to its functional form.

5.2 Query Expansion

In this experiment, we test our metrics on the query expansion task. Briefly, in the query expansion task, we seek to find a small number of terms that can help us to improve the retrieval of relevant documents if they are added to the original query. Here, we enrich a given query with the terms considered close to it according to the resistance kernel.

Datasets: We use four TREC datasets[2] to analyze the performance of our method on the query expansion task: Genomic Track 2003, Genomic Track 2004, Ad hoc TREC 7, Ad hoc TREC 8. The key properties of these datasets are summarized in Table 3. Each TREC dataset comes with 50 test queries and the list of relevant documents assessed by human experts for each query.

TREC	**Type**	**# of docs**	**n**	**m**	**Term type**
Genomic-03	abs	500k	349K	5K	gene/protein
Genomic-04	abs	4.5mil	1123K	5K	gene/protein
Ad Hoc 7	doc	550k	469K	20K	words
Ad Hoc 8	doc	550k	469K	20K	words

Table 3: TREC datasets used in for query expansion (abs=abstract, doc=document, n = total # of terms, m = # of terms used)

Experimental setup: Since there are no query expansion baselines, we use our methods in combination with Terrier search engine [3] to rank the documents and observe its relative performance to the baselines. Terrier is a search engine that parses and indexes the document corpus to build its own vocabulary; its performance can be further improved by doing query expansion first. For baselines, we use: (1) Terrier search engine without query expansion, and (2) Terrier with the PRF-based (pseudo-relevance feedback) [Xu and Croft, 1996] query expansion. PRF methods are the state-of-the-art methods for query expansion that use auxiliary searches to expand original queries. They use all terms in the term vocabulary. We report the best results from the DFR-Bo1 model included in Terrier, which is based on Bose-Einstein statistics [Macdonald *et al.*, 2005]. In contrary to PRF, to run our methods on Ad Hoc 7 and 8 datasets efficiently, we subsample and work with 25% of the overall terms. Yet the end results are very comparable to those of PRF.

Results: Table 4 summarizes the result for the experiment. The statistics used in the evaluation is a widely used document retrieval evaluation metric, the Mean Average Precision (MAP) [Buckley and Voorhees, 2005]. The table shows that our kernel-based query expansion either outperforms or comes close to Terrier's PRF-based expansion baseline which is the state-of-the-art. The difference in Ad Hoc 8 can be explained by applying our method on a reduced term space which includes approximately 25% of the original terms.

6 Conclusions

In this paper, we developed a graph-based framework for constructing text metrics to compare any two arbitrary text components. One important feature of our framework is being *global* meaning that as opposed to the traditional document similarity metrics, the metrics produced by our framework are able to detect the relevance between two text components for which their corresponding terms neither overlap nor co-occur in the same sentence/document across the corpus.

The other key feature of our framework is that it produces a consistent distance measure for two input texts regardless

[2] http://trec.nist.gov/data.html

[3] http://www.terrier.org

Methods	Genomic 03	Genomic 04	Ad Hoc 7	Ad Hoc 8
Terrier	0.19	0.31	0.18	0.24
Terrier+PRF	0.22	**0.37**	**0.22**	**0.26**
Terrier+ $K_{resistance}$	**0.24**	**0.37**	**0.22**	0.25

Table 4: MAP of methods on document retrieval tasks on TREC data

of their sizes (e.g., comparing a term vs. an abstract). We achieved this property by generalizing the distance between two terms to the distance between two sets of terms. To avoid recalculations in computing the distance between arbitrary sets, we proposed an efficient approximate technique which uses the results of one-time spectral decomposition to compute the distance between any two given sets in an online fashion. Moreover, to scale up our framework for large-scale corpora, we developed an approximate subsampling technique which dramatically reduces the order of computations. We experimentally showed that our technique is reasonably robust even if we use moderately small subsamples.

To show the merits of our framework in practice, we used the metrics constructed by our framework for term prediction and query expansion. In both experiments, our metric outperformed the traditional baselines. These very promising results justify further investigation, refinements and possible deployment of our framework for solving real world problems.

Acknowledgments

This research was supported by grants 1R01LM010019-01A1 and 1R01GM088224-01 from NIH. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

[Bach, 2008] F. Bach. Graph kernels between point clouds. In *ICML*, 2008.

[Bordino *et al.*, 2010] I. Bordino, D. Donato C. Castillo, and A. Gionis. Query similarity by projecting the query-flow graph. In *SIGIR '10*, pages 515–522. ACM, 2010.

[Buchman *et al.*, 2009] S. M. Buchman, A. B. Lee, and C. M. Schafer. High-dimensional density estimation via sca: An example in the modelling of hurricane. *Statistical Methodology*, In Press, 2009.

[Buckley and Voorhees, 2005] C. Buckley and E. M. Voorhees. Retrieval system evaluation. *TREC: experiment and evaluation in information retrieval*, 2005.

[Cai *et al.*, 2005] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Trans. Knowl. Data Eng*, 17(12):1624–1637, 2005.

[Chung, 1997] Fan R. K. Chung. *Spectral Graph Theory*. Amer Mathematical Society, May 1997.

[Cohn and Chang, 2000] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML '00*, pages 167–174, July 2000.

[Collins-Thompson and Callan, 2005] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *CIKM '05*, pages 704–711, 2005.

[Dillon *et al.*, 2007] Joshua Dillon, Yi Mao, Guy Lebanon, and Jian Zhang. Statistical translation, heat kernels, and expected distance. In *UAI*, pages 93–100, 2007.

[Doyle and Snell, 1984] P G Doyle and J L Snell. *Random Walks and Electrical Networks*. The Mathematical Association of America, Washington DC, 1984.

[G. Cao and Robertson, 2008] J. Gao G. Cao, J. Y. Nie and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR '08*, pages 243–250, 2008.

[Klein and Randić, 1993] D. J. Klein and M. Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12:81–95, 1993.

[Kondor and Jebara, 2003] R. I. Kondor and T. Jebara. A kernel between sets of vectors. In *ICML*, pages 361–368. AAAI Press, 2003.

[Kondor and Lafferty, 2002] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML*, pages 315–322, 2002.

[Lebanon, 2006] G. Lebanon. Metric learning for text documents. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4):497–508, Apr 2006.

[Macdonald *et al.*, 2005] C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of glasgow at trec 2005: Experiments in terabyte and enterprise tracks with terrier. In *TREC '05*, 2005.

[Moldovan and Rus, 2001] D. Moldovan and V. Rus. Explaining answers with extended wordnet. In *ACL '01*, 2001.

[Ng *et al.*, 2001] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS '01*, pages 849–856, 2001.

[Salton and McGill, 1983] G. Salton and M. J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[Wang and Hauskrecht, 2008] S. Wang and M. Hauskrecht. Improving biomedical document retrieval using domain knowledge. In *SIGIR '08*, pages 785–786. ACM, 2008.

[Wang *et al.*, 2010] S. Wang, M. Hauskrecht, and S. Visweswaran. Candidate gene prioritization using network based probabilistic models. In *AMIA TBI*, 2010.

[Xu and Croft, 1996] J. Xu and B. W. Croft. Query expansion using local and global document analysis. In *SIGIR '96*, pages 4–11. ACM, 1996.

[Zhu *et al.*, 2006] X. Zhu, J. Kandola, J. Lafferty, and Z. Ghahramani. Graph kernels by spectral transforms. *Semi-supervised learning*, pages 277–291, 2006.