RI:Small: Beyond Titles and Abstracts: Text + Links are Better Together

Project Summary: Kwc Version 6 (5/21)

Overview

Opportunity: There has been considerable interest recently in language models (BERT [1]) and graphs (graphical neural nets (GNNs) [2]) for applications in Natural Language and Information Retrieval such as: web search, Academic Search [3, 4], recommendation systems, collaborative filtering and traffic analysis (for the intelligence community). We will focus on Academic Search because of the availability of data [4]. Benchmarks [5] tend to focus on clean snapshots from a few years ago, but it is important to experiment with large (and growing) citation graphs to appreciate network effects (Metcalfe's Law [6]) and timeliness. We need methods to cope with realities such as missing values and bad data. Incremental updates are needed to keep up with growth.

Use Cases: Four uses cases will be considered: (1) recommendations, (2) finding experts [7, 8, 9, 10], (3) routing submissions to reviewers [11, 12] and (4) summarization of sets of documents, emphasizing pairwise similarities between documents (unlike tl;dr methods, which process each document one at a time, missing pairwise relations). Similarity plays a key role in these use cases. There is an opportunity for creative ensembles of deep nets and spectral clustering [13, 14, 15, 16, 17, 18, 19]. Deep nets have been effective for capturing similarities of texts, and spectral clustering is promising for links. Spectral clustering can be viewed as a generalization of PageRank [20] with more than one hidden dimension (Eigenvector). For routing submissions to reviewers, many systems focus on titles and abstracts, but not references. To capture references, we need methods that take advantage of links at inference time, unlike GNNs, which use links for fine-tuning, but not for inference. Since most submissions cite recent literature, it is important to keep our data structures and benchmarks up to date. Web search companies know benchmarks need to be updated frequently because the web is a moving target. So too, timeliness is important for our use cases.

Method: We will take advantage of multiple embeddings including deep nets (**S**) Specter [21, 22, 23] and spectral clustering (**P**) ProNE [19]. **S** cosines are large when papers share similar words and **P** cosines are large when papers are near one another in the citation graph. Combinations of **S** and **P** are robust to realities such as missing values and bad data. Diversity over representations creates opportunities for diverse computing environments. GPUs with GBs of RAM are popular for deep nets; CPUs with TBs of RAM may be preferable for spectral clustering.

We view the literature as a conversation between authors (**S**) and the audience (**P**), somewhat like social media. **S** cosines are time invariant since papers do not change after publication, but **P** cosines evolve as more citations flow in, and new perspectives take hold within the community.

Keywords: Deep Nets; Spectral Clustering; Academic Search; Summarization; Citations

Broader Impacts

The proposed work will make it it easier for everyone to make better use of the scientific literature. There are many practical applications such as topic classification, information retrieval, recommender systems, expertise finding and routing papers to reviewers [24]. This will help us make optimal use of expertise, significantly improving the lives of people in resource-poor countries and environments by, for example, encouraging scientific experts to work on critical projects.

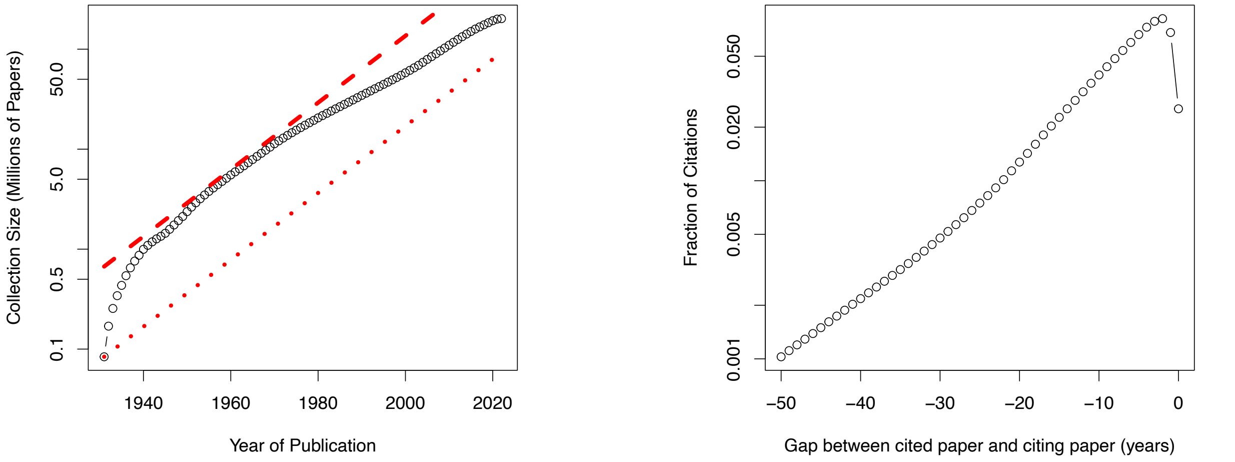
Introduction

There has been considerable interest recently in language models (BERT [1]) and graphs (graphical neural nets (GNNs) [2]) for applications in Natural Language and Information Retrieval such as: web search, Academic Search [3, 4], recommendation systems, collaborative filtering and traffic analysis (for the intelligence community).

We will focus on Academic Search because of the availability of data. Semantic Scholar [4] supports bulk downloads and ad hoc queries for many fields in their databases including titles, abstracts, authors and references. This data makes it possible to construct a citation graph of $N \approx 200M$ papers (nodes) and $E \approx 2B$ citations (edges). (We will use k for thousands, M for millions and B for billions.)

The citation graph can be partitioned along the time dimension, so that it is possible to roll the graph back in time. In this way, we can older views of the graph to predict more recent views of the graph. Benchmarks such as the 2021 KDD-Cup [5] tend to focus on clean snapshots from a single point in time (from a few years ago), but it is important to experiment with large (and growing) citation graphs to appreciate volume and velocity, as illustrated in Figure 1.

According to [25], "global scientific output doubles every nine years." The observed black points fall between the two red lines, suggesting the doubling rate was even faster during the post war years (before 1960). It is perhaps even more impressive to see exponential growth in recent decades, now that the raw numbers are so much larger than they used to be.



(a) 200M papers in [4]: Exponential growth.

(b) Most citations refer to recent publications.

Figure 1: Red lines double every 9 years [25] (with different initial conditions).

Time Invariance And Incremental Updates

It is natural to model a paper as cast in stone when it is published, but we prefer to view the literature as a conversation, somewhat like social media. For example, while Turing's paper [26] has not changed since it was published in 1936, there is a large (and growing) body of work that builds on his contribution. The value of a paper to society is a combination of the primary source plus audience appreciation (subsequent literature and secondary sources).

Some embeddings, M, evolve over time, and some do not. The value of a paper combines time-invariant contributions from authors (Specter embeddings of abstracts, M_S), with contributions from the audience (proposed embeddings, M_P, based on citation graph). Abstracts do not change after publication unlike citations that accumulate over time. Thus, M_S is time invariant, unlike M_P. In this project, we would like to study how relevance and impact evolve over time.

Some of these embeddings take a long time to compute from scratch. Given G's rapid growth (as shown in Figure 1), it is highly desirable to support incremental updates. One suggestion for incremental updates is to compute embeddings for overlapping sets. Suppose we have a set of papers published before t_1, and a set of recent updates published after t_2. Let M_1 and M_2 be embeddings for the two sets. We then need a method to align vectors in M_1 with M_2.

One approach which is probably too simple to work is the Orthogonal Procrustes Problem [27, 28]. Suppose we construct M_1 and M_2 to have a large overlap (where $t_2 \ll t_1$), then we can select rows from M_1 and M_2 for the overlapping papers. Let O_1 and O_2 be the vectors for the overlap. Use the Orthogonal Procrustes Problem to find an $R \in \mathbb{R}^{K \times K}$ that minimizes $|O_1 - O_2 R|$. In this way, we can support incremental updates. That is, we can compute M_2 without recomputing M_1.

Of course, it is likely that we will need to consider more flexible solutions. Rotations may be more successful in small local neighborhoods. In this way, local rotations can be viewed as a kind of piece-wise linear interpolation. If this does not work out, it may be necessary to consider more sophisticated methods in machine learning.

Metcalfe's Law And Network Effects

In addition to volume and velocity, it is also important to experiment with large graphs in order to appreciate network effects (Metcalfe's Law [6]). Metcalfe's Law dates back to the early days of 3Comm. In 1979, they were selling networks with three devices, $n = 3$, two personal computers connected to a single printer. Metcalfe argued that they should sell larger networks (and more 3Comm products) because costs scale with n and benefits scale with n^2. Businesses do well when network effects create economies of scale. Network effects are important for telephone companies and internet businesses such as Google, LinkedIn, Facebook, Twitter, and Snapchat, as Metcalfe argues in a survey of 40 years of Ethernet [29]. We believe that network effects are also important for Academic Search. Four use cases will be considered:

1. recommendations,
2. finding experts [7, 8, 9, 10],
3. routing submissions to reviewers [11, 12] and
4. summarization of sets of documents, emphasizing pairwise similarities between documents (unlike tl;dr methods, which process each document one at a time, missing pairwise relations).

Similarity plays a key role in these use cases. There is an opportunity for creative ensembles of deep nets and spectral clustering [15, 16, 19]. Deep nets have been effective for capturing similarities of texts, and spectral clustering is promising for links. Tables 1-2 compare two estimates of similarity, cos_S and cos_P. The former is based on Specter [21], a deep net based on SciBERT [30], and fine-tuned with citations. The latter is a spectral clustering of the citation graph. Documents with large cos_S have similar abstracts, and documents with large cos_P are near one another in the citation graph.

Use Case 1: Recommendation Systems

It is becoming increasingly difficult to keep up with the literature. Suppose you found one paper, and you suspect there are more because the paper is in a "hot" area. Can we use the one paper as a query, q, to find more? Tables 1-2 find more papers in two "hot" areas, starting with two different queries, q:

- Table 1, q =[31]: "DeepWalk: online learning of social representations" [15]
- Table 2, q = [32]: "On the Dangers of Stochastic Parrots..." [33]

f	$cos_S(q,c)$	$cos_P(q,c)$	**Cites**	**Paper**
f_S	0.984	0.827	0	Learning of Social Representations [34]
f_S	0.809	0.951	2	Topic-aware latent models for representation learning on networks [35]
f_S	0.797	0.947	4	SimWalk: Learning network latent representations with social relation similarity [36]
f_S	0.783	0.033	0	Deep Representation Learning on Complex Graphs [37]
f_P	0.771	0.999	6007	node2vec: Scalable Feature Learning for Networks [38]
f_P	0.711	0.998	3632	LINE: Large-scale Information Network Embedding [39]
f_P	0.664	0.997	1025	A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications [40]
f_P	0.711	0.996	1157	metapath2vec: Scalable Representation Learning for Heterogeneous Networks [41]

Table 1: Approximate n-best matches for query, q = [31]: "DeepWalk..." [15]. Four candidates, c, are shown for f_S (Specter), followed by four for f_P (Proposed). The latter have more citations.

f	**Rank**	$cos_S(q,c)$	$cos_P(q,c)$	**Citations**	**Paper**
f_S	1	0.794	0.957	3	What Language Model to Train if You Have One Million GPU Hours? [42]
f_S	2	0.779	0.976	117	On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines [43]
f_S	3	0.777	0.961	1	Emergent Properties of Finetuned Language Representation Models [44]
f_P	1	0.615	0.992	139	RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models [45]
f_P	2	0.750	0.992	49	The Radicalization Risks of GPT-3 and Advanced Neural Language Models [46]
f_P	3	0.542	0.992	52	Persistent Anti-Muslim Bias in Large Language Models [47]

Table 2: Approximate n-best matches for query, q = [32]: "On the Dangers of Stochastic Parrots..." [33], a paper on Responsible AI aspects of deep nets. All six candidates are about deep nets, but f_P candidates have more citations, and they are more about toxicity and Responsible AI.

The two tables compare cos_S and cos_P. Both methods use approximate nearest neighbors (ANN) [48, 49] to find recommendations with large cosine scores. The recommendations above the line have large cos_S, whereas the recommendations below the line have large cos_P. The tables show both cosine scores for all recommendations so one can see appreciate the similarities and differences. This redundancy will be used to deal with realities such as missing values and bad data.

There is a considerable literature on definitions of relevance [50, 51, 52]. Many recommendation systems focus on "relevance" (overlap in words between queries and candidates), but in addition to that, we should also consider credibility. The recommendations in Tables 1-2 are all on deep nets, but the proposed method recommends papers with more citations (impact).

Most papers are not worth reading. We should not recommend a paper merely because it is buzz-word compliant. Most papers are not cited much, if at all. If possible, we should recommend papers based on behavioral signals (estimates of priors such as citations, page views and downloads). This work will focus on citations because that citations are more available and less sensitive than other behavioral signals, and consequently, citations are easier to share and easier to discuss in the academic literature. That said, we believe that results based on citations may have consequences for enterprises in industry and defense with access to more sensitive signals.

The query system illustrated in Tables 1-2 can not only be used for IR tasks, but we believe that it can also be useful for NLP tasks such as topic modeling and document summarization. Documents near the query tend to be on similar topics. This may not be surprising with Specter embeddings, f_S, given the literature on combinations of BERT and topic modeling [53, 54, 55, 56]. But it appears that the proposed method is also finding documents on similar topics. Thus, if one is interested in studying the use of text and other NLP features to study NLP applications such as topic modeling and summarization, one can use the proposed method to create training materials. While constructing materials to study NLP questions, it is desirable to avoid NLP features in the development of the training materials. The proposed method uses features from the citation graph instead of using NLP features, and therefore, the proposed method is more suitable for constructing training materials to study topic modeling.

More Use Cases

Suppose you are writing a paper and you just made an assertion that should be backed up with a citation. Could you highlight the assertion and ask the system to suggest some candidate citations? Is this process invertible? Suppose we start with a set of papers. Can the system suggest some assertions that calls out similarities and differences among the papers?

Suppose you are writing a paper and you suspect you may be missing some important references. Could you highlight a few references and ask the system to suggest some more? It is likely that the system will find too many relevant papers. How do you know where to begin? We could use a method for summarizing a large set of papers. How are this year's ACL papers differ from last year's? Similarly, compare and contrast papers by the Bengio brothers. How do Sammy's papers differ from his brother's? Many systems for summarizing documents process each paper, one by one. Suppose we want to identify clusters of papers, and compare and contrast similarities and differences among clusters.

Finding references is like finding experts. In large enterprises, it is hard to know who knows what. A number of systems have been built where the answer is a contact instead of a document [7, 9]. Routing submissions to reviewers [11, 12] is related to finding experts.

Would it be possible for the system to write much of the "related work" section automatically? Ideally, after reading a well-written "related work" section, the reader should know where the paper is going, and what the contribution will be. It may be asking too much of the system to anticipate the punch line, but we should be able to build a system that will return a comprehensive set of related papers. The system could also cluster the papers into topics, and suggest some ways to organize the literature into a larger structure such as a timeline, a set of topics, schools of thought, directions forward, etc.

Systems of this kind could also help reviewers. Reviewers are often asked about missing references. We could even imagine our project creating a set of tools to be used by both authors and reviewers to improve their work, and to prepare for feedback from the other. If authors know that reviewers will use these tools, then they should run them first. And if reviewers know that authors have already run these tools in the more obvious ways, reviewers should seek more creative ways to run these tools.

One could think of queries as prompts. It has become popular recently to engineer prompts for ChatGPT [57]. So too, one could engineer queries for recommendation systems.

Use Cases 2-3: Finding Experts And Reviewers

$$f(i) \approx \sum_{j \in (i,\cdot)} f(j) \tag{1}$$

These two use cases are similar to case 1 above, except that candidates should be people (experts/reviewers) as opposed to papers. There are some interesting challenges for mapping between people and vectors. Perhaps the simplest starting point is to represent authors by the centroids of their papers.

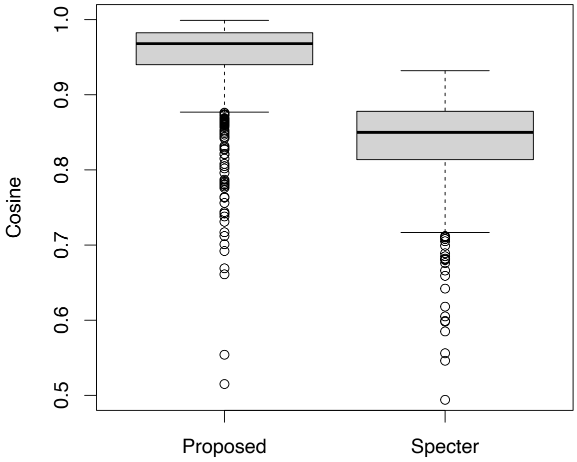


Figure 2: Vectors for a papers are close to the centroid of vectors for their references.

In some use cases (use case 1 and 2 for recommending papers and finding experts), we can assume the queries are members of the documents we have downloaded from [4], but this assumption is not appropriate for assigning submissions to reviewers (case 3), because submissions are unlikely to be in the document collection of (mostly) published papers. For finding reviewers, let us assume the query is a submission, which contains a number of potentially useful fields such as: a title, abstract, authors, body and references. Previous work focuses on titles and authors, though we believe that references are also useful, perhaps even more so than titles and abstracts. The author field is probably also useful, though it may be inappropriate to use that, especially for conferences with double blind reviewing. We hope to find ways to take advantage of additional features such as the body of the submission.

Applying Specter to submissions is relatively straightforward; we can follow the suggestion in [58] and simply apply the Specter model on HuggingFace to titles and abstracts. But how do we use the proposed method to map submissions to vectors? Thus far, we have applied the proposed method to papers in the citation graph. How do we estimate vectors for submissions, assuming submissions are not in the graph? Eqn (1) suggests a simple way forward. That is, the vector for paper i is close the the centroid of i's references. We hope to improve over the centroid approximation, but it is a reasonable place to start.

Figure 2 tests the centroid approximation on a set of 983 papers. For each of these, we have $v = f(i)$, and we obtain, $\hat{v}$ by applying Eqn (1) to the references of i. The boxplots in Figure 2 show

	Citation				**Applicable**	**Method(s)**
Abstracts	**Graph**	**Coverage**	**Papers**	**Region**	**Specter** (f_S)	Proposed (f_P)
✓	✓	31%	65M	tan	✓	✓
✓	✗	16%	34M	red	✓	✗
✗	✓	22%	46M	green	✗	✓
✗	✗	30%	63M	white	✗	✗
		100%	208M			

Table 3: Data for Figure 3; adding proposed method expands coverage into green region.

$cos(v, \hat{v})$. The cosines are large, especially for the propose method. These large cosines suggest the centroid approximation is often effective, especially for the proposed method. Even for Specter, it may be helpful to smooth the vectors obtained from the submission's abstract with vectors obtained from the references. GNNs typically use links for fine-tuning, but there may also be an opportunity to use links at inference time, as well.

In general, we expect short-term weather forecasting to be easier than long-term weather forecasting. That is, it is easier to predict whether it will rain tomorrow than a week from now. So too, we would expect the cosines in Figure 2 to degrade with time. That is, we believe it is important to keep our vectors up to date. We expect the centroid approximation to become less effective as time passes between the publication of references, j, and submissions, i. The experiment in Figure 2 establishes plausibility, but in that case, we did not control for the time between i and j. We plan to investigate how important it is to keep our vectors up to date by measuring empirically the effectiveness of Eqn (1) as a function of the time between i and j.

Coping With Realities: Missing Values And Bad Data

Multiple representations introduce possibilities for dealing with realities such as missing values, as illustrated in Table 3 and Figure 3. Specter is applicable in the red region but not the green, whereas the proposed method is applicable in the green region but not the red. The combination has more coverage than either by itself.

Multiple representations also create possibilities for detecting bad data. The data from Semantic Scholar [4] is extremely useful, but it is not free from errors. Bad data is often associated with large differences between cos_P and cos_S. In Table 1, cos_P is too low for candidate [37] because of gaps in the citation graph. Table 4 shows some examples where cos_S is too high because of bad data in the data structure for abstracts.

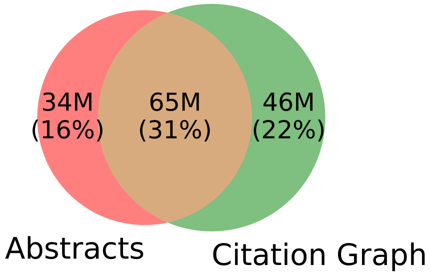


Figure 3: Venn diagram of [4]

Bad Data In Field For Abstracts

In Table 4, we illustrate an opportunity to take advantage of multiple perspectives to detect errors. Most of the abstracts in Semantic Scholar are correct, but there are a few documents in Semantic Scholar where the abstract field in the database was incorrectly replaced with some boilerplate from JSTOR such as:

> *JSTOR is a not-for-profit service that helps scholars, researchers, and students discover,*

use, and build upon a wide range of content in a trusted digital archive...

There are similar errors for a number of papers in the data downloaded from [4]. When this error happens to the query, then the recommender system illustrated in Tables 1-2 is likely to find a number of candidates with similar errors. Consider Table 4. In this case, the abstract field for the query [59] and the candidates [60, 61, 62] are all replaced with similar boilerplate from JSTOR. These errors produce large (misleading) cos_S because the values in the abstract fields of the database are all similar, though obviously, the papers have little in common.

f	**Rank**	cos_S	cos_P	**Citations**	**Paper**
f_S	7	0.777	0.097	15	The Value of Natural Sounds [60]
f_S	8	0.766	0.089	16	Re-Examining the Foundations [61]
f_S	9	0.757	0.106	8	Facts and Faith in Biblical History [62]

Table 4: Multiple embeddings create opportunities for error detection. Normally, $cos_S \approx cos_P$, but errors in the abstract database can produce inflated estimates of cos_S. Errors can often be detected by comparing cos_S and cos_P, because different errors impact different cosines differently.

Fortunately, these errors can be detected by comparing cos_S with cos_P, because different errors impact different cosines differently; cos_S is sensitive to errors in abstracts whereas cos_P is sensitive to errors in the citation graph. Multiple redundant estimates of similarity create opportunities for robustness.

Notation

We will use the following notation in the rest of this proposal:

1. Units: k is for thousands, M is for millions, B is for billions. Similarly, GB refers to gigabytes and TB refers to terabytes.
2. Let $N \approx 200$M be the number of documents in the collection.
3. Let $E \approx 2$B be the number of citations in the collection.
4. Let K be the number of hidden dimensions (typically, $K = 768$ for models based on BERT, and $K = 280$ for models based on Spectral Clustering and Linear Algebra).
5. Let f be a model and M be an embedding. Models f can be applied at inference time to novel inputs, or they can be rows of M. Embeddings, M, are large matrices: $M \in \mathbb{R}^{N \times K}$, where $f(d_i)$ is the row in M for document d_i. That is, $f(d_i) = M[d_i,]$.
6. Let f^{-1} be the inverse of f. f maps document ids to vectors ($f(d_i) = M[d_i,]$), and f^{-1} maps vectors to document ids ($f^{-1}(M[d_i,]) = d_i$). We can implement f^{-1} with approximate nearest neighbors (ANN) [63, 48, 49], so f^{-1} can be applied to vectors that do not match any of the rows in M exactly.
7. The subscripts S and P will be used as necessary to distinguish f_S from f_P, and M_S from M_P. The subscript S refers to Specter, and the subscript P refers to the model proposed here, which is based on ProNE [19]. More subscripts will be introduced for additional embeddings.

Multiple Representations

This proposal will make use of both text and citations at inference time. A number of systems such as Specter and Graphical Neural Nets (GNNs) [2] take advantage of citations when fine-tuning, but not at inference time. The goal of these fine-tuning processes is to produce a model, f, that will be applied at inference time to a string, s, typically titles and abstracts.

We are concerned that this approach misses the opportunity to query on combinations of text and links at inference time. For example, if one wanted to route a submission to qualified reviewer that is familiar with the relevant background literature, we suggest the routing system should take advantage of more than the text in the abstract. In particular, the references in the submission are probably more helpful for identifying the relevant background literature. When we started EMNLP, we often sent submissions to a reviewer that was cited in the submission. The authors of the background literature likely to be well-informed and sympathetic with the approach.

In this work, we generalize f to take advantage of multiple representations (and additional properties) so estimates of similarities will succeed even when abstracts (and other properties) are corrupted and/or missing at inference time. That is, we assume document ids, d, are associated with various properties that are often available (but not always):

- text: titles, abstracts, tl;dr (too long; didn't read) summaries, full text, and
- context (links): citations, citing sentences, plus
- more: authors, venues, fields of study

In this way, the proposed approach:

1. expands coverage into the green region (with missing abstracts),
2. enables error detection, which is important when abstracts are corrupted (Table 4), and
3. enables queries over strings (titles and abstracts), links (citations) and more.

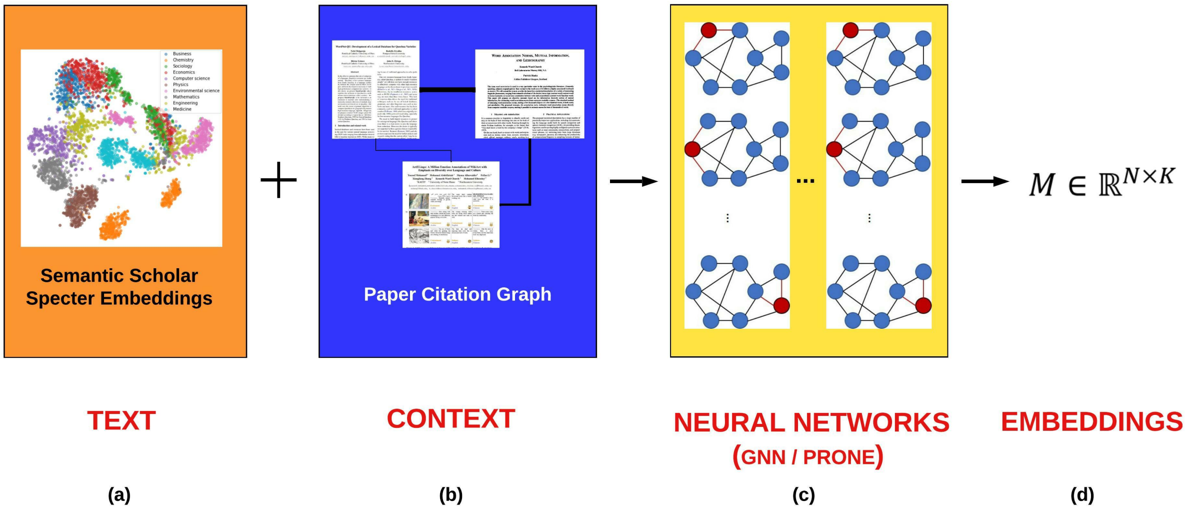


Figure 4: Embeddings are typically based on text (titles and abstracts). We will create multiple embeddings based on combinations of text and context (links such as citations and citing sentences).

Our project is intended to show that multiple representations of documents are better together, as illustrated in Figure 4. The next section will introduce the Cos-Dist assumption, where cosines in embeddings can be used to estimate distances in graphs, and vice versa. We view embeddings, M, and graphs, G, as different representations of documents, somewhat analogous to frequency and time domain representations in speech. Operations such as filtering can be implemented as multiplication in the frequency domain, or convolution in the time domain. So too, under Cos-Dist, one can estimate document similarities with cosines in the embedding space and distances in graphs. Different representations have different advantages and disadvantages. Embeddings are convenient for estimating cosines, and finding approximate nearest neighbors, though graphs are more compact.

Graphs are more compact because their size depends on E (nonzero edges), unlike embeddings, which are stored as dense matrices, $M \in \mathbb{R}^{N \times K}$. Graphs are more compact because $K \gg E/N$. Recall that $K \gg 100$ and $E/N \approx 10$. We will refer to E/N as the average citation rate (about 10 because $N \approx 200$M papers and $E \approx 2$B citations).

More Embeddings

We will experiment with many embeddings, in addition to M_S and M_P. One such embedding is based on citing sentences [64, 65, 66], which are like anchor text [67] in web search. We expect citing sentences to be helpful when subsequent literature introduces contemporary terminology. Consider, for example, Turing's seminal paper [26], which introduced what is now known as a "Turing Machine," though of course, Turing did not use that term in his paper. The term is very common in sentences that cite his paper. About one-third of them mention Turing Machines. More generally, it is common for the subsequent literature to name important concepts after important figures in the literature, but it is rare to find the modern terminology in the primary source. We have already run Specter on a billion citing sentences, and hope to show that citing sentences are helpful. Citing sentences might be even more useful than primary sources for appreciating important contributions in contemporary contexts.

In addition to citing sentences, there are a number of alternatives to Specter in the literature including Specter2 [23], Link BERT [68] and SciNCL [69]. The Semantic Scholar API provides convenient access to Specter1 embeddings [22], but their API does not currently support these alternatives. We will run these alternative models on as many papers as possible and distribute the results on Globus [70]. We will also distribute indexes for approximate nearest neighbor (ANN) search.

Note that these files are large. Embeddings are dense matrices, $M \in \mathbb{R}^{N \times K}$, where $N \approx 200$M and $K \approx 768$. If each value in M is a 4-byte float, then each embedding is 614GBs. The indices are relatively small, though probably too large for GitHub. Each index is a permutation of N (about a GB). We often use about a dozen indexes per embedding.

Cos-Dist: Multiple Views Of Similarity

Prior work tends to encode documents as vectors in just one way. We propose multiple embeddings, f_1, f_2,..., to capture multiple perspectives such as titles, abstracts, full text and citations. Let $cos(f(i), f(j))$ denote the similarity of i and j, where i, j can be strings, documents and/or "topics" [71, 72, 73, 74]. Cosines of vectors based on text (e.g., bags of words [75, 76, 77], BERT [1, 21, 30]) denote word similarity, whereas cosines based on node2vec/ProNE [78, 19] (methods involving spectral clustering of citation graph, G) can be interpreted in terms of $dist(i, j)$, distances in G.

We observe that documents that are similar in at least one way, tend to be similar in other ways, as well. We refer to this assumption as Cos-Dist: $cos(f_1(i), f_1(j)) \sim cos(f_2(i), f_2(j)) \sim dist(i, j)$. Diversity over representations opens many opportunities:

1. Interpretability (Cos-Dist): Similarity of documents can be estimated as cosines in embeddings, $cos(f(i), f(j))$, or distances in G. In other words, embeddings can be viewed as an alternative representation of graphs. Both representations have advantages and disadvantages. Graphs are compact, but embeddings are convenient for computing cosines and ANNs.

2. Redundancy: if we have two embeddings, f_1 and f_2, then we have three redundant estimates of document similarity: (a) $cos(f_1(i), f_1(j))$, (b) $cos(f_2(i), f_2(j))$ and (c) distances in G. Redundant estimates create opportunities for error detection and error correction.
3. Cos-Dist holds for embeddings based on Linear Algebra as well as embeddings based on nonlinear deep nets, creating opportunities to generalize results from relatively well understood Linear Algebra to recent advances in deep nets, as suggested in [18].
4. Diversity over representations introduces opportunities for diverse computing environments. For example, it has been popular to use GPUs with GigaBytes (GBs) of RAM for deep nets, but CPUs with TeraBytes (TBs) of RAM may be preferable for embeddings based on Linear Algebra. When we have TBs of RAM, it becomes feasible to compute the SVD of a large citation graph in RAM, as we will do in the discussion of ProNE and prefactorization. Standard recipes for training deep nets with mini-batches can be viewed as an external memory algorithm. Because external memory is slower than RAM, CPUs with TBs may be preferable to GPUs with GBs, at least for some memory-bound computations.

Figure 5 provides some evidence for Cos-Dist, showing that distances in the citation graph are related to cosines in embeddings. We created pairs of documents with a random walk. We started with nearly 1M document ids, selected at random. From each of these, we randomly selected one of its references. From there, we continued the walk by randomly selecting one of its references, and so on. For each of these pairs, when the cosine is available, we report the length of the shortest path on the x-axis, and cos_S (and cos_P) on the y-axis. More than 20% of the pairs are omitted from the plot for Specter because those cosines are often unavailable (due to missing abstracts).

For both Specter and the proposed method, we find that cosines are negatively correlated with path lengths, though the pattern is stronger for f_P because that method uses an optimization criterion that is closer to Cos-Dist. We believe, however, that Cos-Dist holds for most reasonable embeddings of documents because they are all estimating similarities of the same documents.

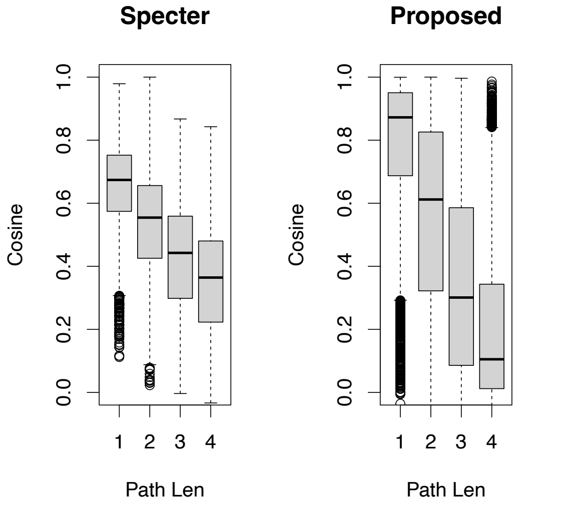


Figure 5: *Cos-Dist*: Cosines in embeddings can be interpreted as the length of shortest path in G: $cos(f(i), f(j)) \sim pathlen(i, j)$.

ProNE

The ProNE method, f_P, is based on the citation graph, G. We start by downloading G from [4]. G is stored as an adjacency matrix, an $N \times N$ Boolean matrix, $M_G \in \{0, 1\}^{N \times N}$, with $N \approx$ 200M nodes (papers). If (and only if) paper d_i cites paper d_j, then $M_G[d_i, d_j] = 1$. This matrix is sparse because there are only $E =$ 2B citations (nonzero edges). G might appear to be a huge matrix but it requires just 7.8 GBs when stored in scipy [79] as an npz file because G is extremely sparse.

We then run the ProNE [19] method in nodevectors [80] to produce an embedding, $M_P \in \mathbb{R}^{N \times K}$, where N is the number of nodes (papers) in G, and K is the number of hidden dimensions. ProNE is a variant of node2vec [78]. Unlike Specter and many variants of node2vec, ProNE is based on

Linear Algebra and spectral clustering, and does not use deep nets.

The first step in ProNE is called *prefactorization*. It uses a memory internal SVD procedure to factor a normalized version of G as UDV^T, where $U, V \in \mathbb{R}^{N \times K}$. While G is sparse, U and V are not. Since we have TBs of RAM, it is feasible to compute this SVD in RAM.

After prefactorization, ProNE uses spectral propagation to compute the embedding M_P so cosines in M_P can be interpreted in terms of distances on G. For efficency, ProNE uses Chebyshev expansion to avoid explicit Eigen decomposition. While the Chebyshev expansion is efficient, it makes a number of copies of large matrices in memory, each of which is the size as U. The total memory requirement is about 5 times larger than U. The output embedding, M_P, is also a dense matrix of the same size as U.

ProNE takes considerable time and space. It takes nearly a week to compute M_P on a CPU with 2 TBs of RAM with $K = 280$. The time complexity is: $O(NK^2 + E)$, but in our case, we can simplify that to $O(NK^2)$ because $NK^2 \gg E$. Increasing K improves the approximation that cosines on M are related to distances on G. Of course, increasing K also increases computational costs (time and space). One of the work items for this proposal is to make it easier to compute M_P:

1. Improve time and space constants,
2. Increase K, and evaluate benefits and costs of doing so, and
3. Provide support for incremental updates to keep up with exponential growth in Figure 1.

Diversity over representations creates opportunities for diverse computing environments. It has been popular to use GPUs with GigaBytes (GBs) of RAM for deep nets (such as BERT), but CPUs with TeraBytes (TBs) of RAM may be preferable for embeddings based on Linear Algebra (such as ProNE). We typically train deep nets (such as GNNs) by grouping edges into mini-batches. Each mini-batch updates model parameters in GPUs. This process is repeated for a few epochs, and therefore, time complexity grows linearly with the size of training set. With ProNE, we run the SVD in TBs of RAM. TBs of RAM costs about the same as GPUs, but TBs of RAM make it feasible to consider algorithms beyond linear time. With sufficiently large memories, it becomes feasible to sort the training data, or to run SVD on it. Many methods in Linear Algebra require more than linear time.

Use Case 4: Document Summarization

Tables 1-2 suggest another NLP application: summarization. In addition to titles (as in Tables 1-2), it would be desirable to produce snippets/summaries to help users choose among the various document recommendations. It is common for summarization systems to process each document one at a time, though it would be desirable for summaries to compare and contrast the choices to help users appreciate similarities and differences between various options. One of the work items for this proposal is to produce summaries that improve over two baselines: (1) tl;dr from Semantic Scholar [4], and (2) summaries produced by ChatGPT [57].

Table 5 shows some tl;dr summaries from the Semantic Scholar API [4] for three candidates from Table 2. These summaries tend to be extracts from abstracts. To illustrate baseline (2), we ran ChatGPT on the text in Table 5 and it produced the following output:

> *The recent research on the dangers of large language models (LLMs) highlights the importance of considering their potential negative impacts on society. As language models become increasingly sophisticated, there is a risk of them degenerating into toxic text or*

Title	**tl;dr (too long; didn't read)**
On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? [32]	Recommendations including weighing the environmental and financial costs first, investing resources into curating and carefully documenting datasets rather than ingesting everything on the web, and carrying out pre-development exercises evaluating how the planned approach fits into research and development goals and supports stakeholder values are provided.
RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models [45]	It is found that pretrained LMs can degenerate into toxic text even from seemingly innocuous prompts, and empirically assess several controllable generation methods find that while data- or compute-intensive methods are more effective at steering away from toxicity than simpler solutions, no current method is failsafe against neural toxic degeneration.
The Radicalization Risks of GPT-3 and Advanced Neural Language Models [46]	GPT-3 demonstrates significant improvement over its predecessor, GPT-2, in generating extremist texts and its strength in generating text that accurately emulates interactive, informational, and influential content that could be utilized for radicalizing individuals into violent far-right extremist ideologies and behaviors.
Persistent Anti-Muslim Bias in Large Language Models [47]	GPT-3, a state-of-the-art contextual language model, captures persistent Muslim-violence bias, demonstrating that it appears consistently and creatively in different uses of the model and that it is severe even compared to biases about other religious groups.

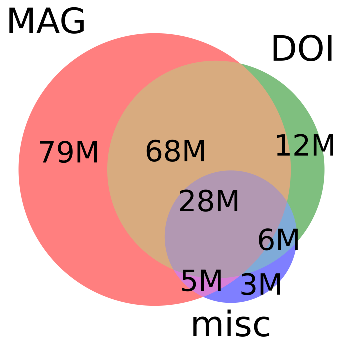
Table 5: tl;dr summaries from Semantic Scholar [4] for some papers from Table 2.

> *perpetuating biases and harmful ideologies. Therefore, it is crucial to evaluate the risks and benefits of LLMs before developing and deploying them.*
>
> *One recommendation for addressing the risks of LLMs is to weigh the environmental and financial costs of building and training these models. LLMs are known to require large amounts of energy and computational resources, which can have significant environmental impacts. It is important to consider these costs alongside the potential benefits of using LLMs.*

Neither of these baselines address the desiderata mentioned above. That is, they fail to compare and contrast the choices. Let d_a and d_b be two documents, and $a = f(d_a)$ and $b = f(d_b)$ be their vectors, respectively. We suggest expressing comparisons and contrasts in the vector space as in Eqn (2). We can then use f^{-1} to find nearby documents, and then report properties of those nearby documents such as titles, abstracts, references, citations, etc.

$$\begin{array}{l} compare(a, b) = ARGMIN_{c \in C}\ |c - a| + |c - b| \\ contrast(a, b) = ARGMIN_{c \in C}\ |c - a| - |c - b| \end{array} \tag{2}$$

Contrasts are similar to comparisons except for a sign change. For comparisons, we plan to cast a wide net and let the set of candidates, C, be the set of 200M documents in Semantic Scholar. For contrasts, we plan to limit the set of candidates to documents that are reasonably close to a for some threshold, T. That is, $|c - a| < T$.



1. 182M papers from MAG [81, 82]
2. 114M papers from DOI [83]
3. 35M papers from PubMed [84]
4. 6M papers from DBLP [85]
5. 5M papers from PubMedCentral [86]
6. 2M papers from arXiv [87]
7. 80k papers from ACL Anthology [88]

Figure 6: Seven sources in Semantic Scholar [4].

Data, Applications & Evaluations

For evaluations, we plan to start with benchmarks on GitHub [89, 90] with baseline results reported in [21, 91]. In addition to these benchmarks from Semantic Scholar, there are also a number of influential benchmarks such as OGB (Open Graph Benchmark) [92, 93], which was used in the 2021 KDD-Cup [5].

OGB was designed to compare graph learning algorithms. While much of the work based on that benchmark is very relevant to this proposal, the rules of the benchmark prohibit the use of outside data such as data from Semantic Scholar. In addition, since the focus is on graph learning, and not on practical applications, the benchmark does not publish the mapping of paper ids to titles and abstracts, and the mapping of author identifiers to author names. While one could probably reverse engineer that mapping, doing so would violate the spirit of the benchmark. Unfortunately, without such a mapping, it is not clear how methods based on this benchmark can be used in practical applications.

The literature on graph learning classifies tasks into (a) node-level, (b) link-level and (c) graph-level. The example of (a) in [93] is closely related to tasks of interest for this proposal, unlike the examples of (b) and (c), which are closer to Knowledge Graph Completion and Chemistry. OBG uses MAG240M for (a). MAG240M is based on MAG (Microsoft Academic Graph) [82]. Unfortunately, Microsoft retired MAG at the end of 2021, though MAG has been transferred to a nonprofit, OpenAlex [94]. OpenAlex is making the data available for bulk access. More importantly, they will keep MAG up to date, which is important because the citation graph is expanding rapidly, as discussed in Figure 1.

Figure 6 shows the importance of MAG. Semantic Scholar is based on 7 sources, but the 5 smaller ones (*misc*) contribute just 3M papers that are not already covered by MAG and DOI. Our field tends to focus on ACL Anthology and arXiv, two relatively small sources.

The 200M papers cover a diverse set of fields of study (fos), though with an emphasis on STEM: Medicine (45M), Chemistry (13M), Computer Science (13M), Biology (13M), Materials Science (10M), Engineering (8M), Physics (7M), Psychology (7M), Mathematics (5M), Political Science (4M), Business (4M), Sociology (3M), Geography (3M), Economics (3M), Environmental Science (3M), Geology (3M), History (2M), Art (2M), Philosophy (1M). We have found other sources such as SSRN (Social Science Research Network) [95] to be more useful for Social Studies, Law and Journalism, even though SSRN is relatively small (1M papers). SSRN also makes use of behavioral signals on downloads, which is extremely useful, especially in fields where it is less common for papers to cite one another. Unfortunately, SSRN does not make its data available for bulk download.

MAG240M is based on a citation graph from an older version of MAG. This graph has 122M papers (nodes) with 1.3B citations (edges). MAG240M also makes use of an author graph with 122M authors (nodes) who wrote 386M papers (edges). A relatively small set of papers have labels. About 1.4M of the 122M paper nodes are arXiv papers, annotated with one or more of about 150 arXiv subject areas. The evaluation is a prediction task: predict a subject area label for the arXiv papers. The data is split by publication year: train (arXiv papers published in 2018 or before), validation (arXiv papers published in 2019), test (arXiv papers published in 2020).

There are a few similar tasks in [90] such as: fos (field of study), and MeSH descriptors. These tasks are more attractive because they provide a mapping from numeric label identifiers to meaningful strings. Moreover, there are only 2M arXiv papers, and it is not clear that subject area labels for these papers will generalize well to papers from other sources. There are many more papers in Semantic Scholar with fos labels (148M), and these labels are easier to interpret. The MeSH task is attractive since the National Library of Medicine has been labeling PubMed papers with MeSH terms for many years. PubMed (35M papers) is also considerably larger than arXiv (2M papers).

As for link prediction, there are a couple of tasks in [90] such as cite prediction that are attractive because they generalize to important practical problems. Recommender systems should be able to answer questions such as what should I read and what should I cite. The cite prediction task, as described in [21], provides pairs of document identifiers with titles and abstracts. The pairs are labeled with 1 or 0, indicating whether the first paper of the pair cites the second member of the pair or not. The task is to predict the label.

In fact, we discovered that the task is more like the training material for Specter, where papers are assigned a positive label if they are known to be within the one or two hop neighborhood. Negative labels are assigned to random pairs (which are sometimes within one or two hops). We will report results on the original task, as well with a version with labels using the simpler description of the task in [21],

The proposed work will improve over evaluations above in two respects. First, most of the evaluations above make use of titles and abstracts and little else. We plan to take advantage of additional features (citations, citing sentences and full text) in a way that is robust to missing/corrupted values. In addition, the evaluation should guide the community toward building effective solutions to practical tasks such as:

1. Label Prediction: predict labels such as subject areas, fos (field of study), MeSH.
2. Link Prediction for Recommender systems: what should I read? what should I cite?
3. Search / Rank Retrieval / Relevance Feedback: given a query (text and/or documents with relevance labels): find documents in the collection that are similar to the query.
4. Paper Reviewer Matching: given a paper (text and/or document id), identify the best set of reviewers for the paper, satisfying a variety of constraints including topic expertise, avoiding conflicts of interest, diversity of reviewer institutions, balanced work loads, and topic variety.
5. Summarize collection of documents: compare and contrast clusters (as opposed to summarizing each document in isolation).

Many of these tasks have been discussed above.

Deliverables / Work Items

1. Better access to literature
2. Resources: Many embeddings for many papers; More models to be posted on HuggingFace; Code to be posted on GitHub

3. Summarization methods to compare and contrast across small (and large) collections of documents
4. Support incremental updates to embeddings based on citation graphs
5. Evaluation: Better numbers, as well as better benchmarks
6. Establish that combinations of text and links are better together (than either by itself)
7. Establish that citing sentences are useful
8. Improve methods for assigning papers to reviewers
9. Theory: Unified framework of deep nets and Linear Algebra

References Cited

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[2] anonymous, “Graph neural networks,” https://snap-stanford.github.io/cs224w-notes/machine-learning-with-networks/graph-neural-networks, 2020.

[3] “”google scholar”,” https://scholar.google.com/.

[4] Semantic-Scholar, “Semantic scholar academic graph api: Providing a reliable source of scholarly data for developers,” https://www.semanticscholar.org/product/api, 2017.

[5] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec, “Kdd cup ogb large-scale challenge (ogb-lsc) session 1,” https://www.youtube.com/watch?v=MblxCqwengI, 2021.

[6] P. Fisk, “Metcalfe’s law explains how the value of networks grows exponentially ... exploring the “network effects” of businesses like apple, facebook, trulia and uber,” https://www.peterfisk.com/2020/02/metcalfes-law-explains-how-the-value-of-networks-grow-exponentially-there-are-5-types-of-network-effects/.

[7] L. A. Streeter and K. E. Lochbaum, “Who knows: A system based on automatic representation of semantic structure,” in *User-Oriented Content-Based Text and Image Handling*, 1988, pp. 380–388.

[8] D. Yimam-Seid and A. Kobsa, “Expert-finding systems for organizations: Problem and domain analysis and the demoir approach,” *Journal of Organizational Computing and Electronic Commerce*, vol. 13, no. 1, pp. 1–24, 2003.

[9] M. T. Maybury, “Expert finding systems,” https://www.mitre.org/sites/default/files/pdf/06_1115.pdf, 2006.

[10] Y. Tu, N. Johri, D. Roth, and J. Hockenmaier, “Citation author topic model in expert search,” in *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 1265–1273. [Online]. Available: https://aclanthology.org/C10-2145

[11] D. Mimno and A. McCallum, “Expertise modeling for matching papers with reviewers,” in *Knowledge Discovery and Data Mining*, 2007.

[12] X. Liu, T. Suel, and N. D. Memon, “A robust model for paper reviewer assignment,” in *ACM Conference on Recommender Systems*, 2014.

[13] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[14] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[16] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 891–900.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[18] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2017.

[19] J. Zhang, Y. Dong, Y. Wang, J. Tang, and M. Ding, "Prone: Fast and scalable network representation learning." in *IJCAI*, vol. 19, 2019, pp. 4278–4284.

[20] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[21] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. Weld, "SPECTER: Document-level representation learning using citation-informed transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Online: Association for Computational Linguistics, Jul. 2020, pp. 2270–2282. [Online]. Available: https://aclanthology.org/2020.acl-main.207

[22] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld, "allenai/specter," https://huggingface.co/allenai/specter.

[23] A. Singh, M. D'Arcy, A. Cohan, D. Downey, and S. Feldman, "allenai/specter2," https://huggingface.co/allenai/specter2.

[24] R. Kinney, "Conference Peer Review with the Semantic Scholar API," https://blog.allenai.org/conference-peer-review-with-the-semantic-scholar-api-24ab9fce2324, 2021.

[25] R. Van Noorden, "Global scientific output doubles every nine years," *Nature news blog*, 2014.

[26] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *J. of Math*, vol. 58, no. 345-363, p. 5, 1936.

[27] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems.* OUP Oxford, 2004, vol. 30.

[28] "scipy.linalg.orthogonal_procrustes," https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.orthogonal_procrustes.html.

[29] B. Metcalfe, "Metcalfe's law after 40 years of ethernet," *Computer*, vol. 46, no. 12, pp. 26–31, 2013.

[30] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3615–3620. [Online]. Available: https://aclanthology.org/D19-1371

[31] “Deepwalk: online learning of social representations,” https://www.semanticscholar.org/paper/fff114cbba4f3ba900f33da574283e3de7f26c83.

[32] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, “On the dangers of stochastic parrots: Can language models be too big?” https://www.semanticscholar.org/paper/On-the-Dangers-of-Stochastic/6d9727f1f058614cada3fe296eeebd8ec4fc512a.

[33] ——, “On the dangers of stochastic parrots: Can language models be too big?” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 610–623.

[34] “Learning of social representations,” https://www.semanticscholar.org/paper/93b050f5bf0567a675979cd564cbe66ff9c3a78f.

[35] “Topic-aware latent models for representation learning on networks,” https://www.semanticscholar.org/paper/21ee2cc0bf41c1b74efb6104edd4df73416b46c1.

[36] “Simwalk: Learning network latent representations with social relation similarity,” https://www.semanticscholar.org/paper/e294339b402ce055d5a5198becc35b2dbbd20a9a.

[37] “Deep representation learning on complex graphs,” https://www.semanticscholar.org/paper/bb11bec51c2e069ef0ddba4eb3117c9dbc8a4584.

[38] “node2vec: Scalable feature learning for networks,” https://www.semanticscholar.org/paper/36ee2c8bd605afd48035d15fdc6b8c8842363376.

[39] “LINE: Large-scale information network embedding,” https://www.semanticscholar.org/paper/0834e74304b547c9354b6d7da6fa78ef47a48fa8.

[40] “A comprehensive survey of graph embedding: Problems, techniques, and applications,” https://www.semanticscholar.org/paper/006906b6bbe5c1f378cde9fd86de1ce9e6b131da.

[41] “metapath2vec: Scalable representation learning for heterogeneous networks,” https://www.semanticscholar.org/paper/c0af91371f426ff92117d2ccdadb2032bec23d2c.

[42] T. L. Scao, T. Wang, D. Hesslow, L. Saulnier, S. Bekman, S. Bari, S. R. Biderman, H. ElSahar, N. Muennighoff, J. Phang, O. Press, C. Raffel, V. Sanh, S. Shen, L. Sutawika, J. Tae, Z. X. Yong, J. Launay, and I. Beltagy, “What language model to train if you have one million gpu hours?” https://www.semanticscholar.org/paper/What-Language-Model-to-Train/bb15f3727f827a3cb88b5d3ca48415c09b40a88f.

[43] M. Mosbach, M. Andriushchenko, and D. Klakow, “On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines,” https://www.semanticscholar.org/paper/On-the-Stability/8b9d77d5e52a70af37451d3db3d32781b83ea054.

[44] A. Matton and L. de Oliveira, “Emergent properties of finetuned language representation models,” https://www.semanticscholar.org/paper/Emergent-Properties-of/79fdff5339017ec92b979efa4dff33d21a69b66e.

[45] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, “Realtoxicityprompts: Evaluating neural toxic degeneration in language models,” https://www.semanticscholar.org/paper/RealToxicityPrompts/399e7d8129c60818ee208f236c8dda17e876d21f.

[46] K. McGuffie and A. Newhouse, "The radicalization risks of gpt-3 and advanced neural language models," https://www.semanticscholar.org/paper/The-Radicalization/02fde8bfd9259a4f53316579eb0bf97213559e5c.

[47] A. Abid, M. S. Farooqi, and J. Y. Zou, "Persistent anti-muslim bias in large language models," https://www.semanticscholar.org/paper/Persistent-Anti-Muslim/4c2733d191e347753bb28afa46a1c55c65e085be.

[48] E. Bernhardsson, "Annoy," https://github.com/spotify/annoy, 2013.

[49] F. Research, "Faiss," https://github.com/facebookresearch/faiss, 2019.

[50] S. Mizzaro, "Relevance: The whole history," *Journal of the American society for information science*, vol. 48, no. 9, pp. 810–832, 1997.

[51] P. Borlund, "The concept of relevance in ir," *Journal of the American Society for information Science and Technology*, vol. 54, no. 10, pp. 913–925, 2003.

[52] T. Saracevic, "The notion of relevance in information science," *Everybody knows what relevance is. But what is it really*, 2017.

[53] N. Peinelt, D. Nguyen, and M. Liakata, "tbert: Topic models and bert joining forces for semantic similarity detection," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 7047–7055.

[54] M. Grootendorst, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," *arXiv preprint arXiv:2203.05794*, 2022.

[55] "BERTopic," https://maartengr.github.io/BERTopic/index.html.

[56] "Topic modeling bert+lda," https://www.kaggle.com/code/dskswu/topic-modeling-bert-lda.

[57] https://chat.openai.com/.

[58] R. Kinney, "Conference peer review with the semantic scholar api," https://blog.allenai.org/conference-peer-review-with-the-semantic-scholar-api-24ab9fce2324, 2021.

[59] M. Liberman and A. Prince, "On stress and linguistic rhythm," https://www.semanticscholar.org/paper/On-stress/b8ec853894551c0e7a822df50dc04eccd613d46f.

[60] J. A. Fisher, "The value of natural sounds," https://www.semanticscholar.org/paper/The-Value-of-Natural-Sounds-Fisher/f3044464aae6d74ef8b7c9a85810b5164e837378.

[61] F. V. Winnett, "Re-examining the foundations," https://www.semanticscholar.org/paper/Re-Examining-the-Foundations/52e0d57cccd3ffc9097ac9ee95c1a2214fdc1c7a.

[62] R. H. Pfeiffer, "Facts and faith in biblical history," https://www.semanticscholar.org/paper/Facts-and-Faith-in-Biblical-History-Pfeiffer/639c1e93818b157541bf5522a7cb2cf564119479.

[63] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604–613.

[64] P. I. Nakov, A. S. Schwartz, M. Hearst *et al.*, "Citances: Citation sentences for semantic analysis of bioscience text," in *Proceedings of the SIGIR*, vol. 4. Citeseer, 2004, pp. 81–88.

[65] V. Qazvinian and D. Radev, "Identifying non-explicit citing sentences for citation-based summarization." in *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, pp. 555–564.

[66] A. Abu-Jbara and D. Radev, "Reference scope identification in citing sentences," in *Proceedings of the 2012 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*, 2012, pp. 80–90.

[67] N. Craswell, D. Hawking, and S. Robertson, "Effective site finding using link anchor information," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 250–257.

[68] M. Yasunaga, J. Leskovec, and P. Liang, "LinkBERT: Pretraining language models with document links," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8003–8016. [Online]. Available: https://aclanthology.org/2022.acl-long.551

[69] M. Ostendorff, N. Rethmeier, I. Augenstein, B. Gipp, and G. Rehm, "Neighborhood contrastive learning for scientific document representations with citation embeddings," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11 670–11 688. [Online]. Available: https://aclanthology.org/2022.emnlp-main.802

[70] "Globus," https://www.globus.org/.

[71] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–480, 1992. [Online]. Available: https://www.aclweb.org/anthology/J92-4003

[72] D. M. Blei, "Probabilistic topic models," *IEEE Signal Processing Magazine*, vol. 27, pp. 55–65, 2010.

[73] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.29

[74] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," *arXiv preprint arXiv:1207.4169*, 2012.

[75] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[76] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[77] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[78] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[79] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[80] "Nodevectors," https://github.com/VHRanger/nodevectors.

[81] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia, "Microsoft academic graph: When experts are not enough," *Quantitative Science Studies*, vol. 1, no. 1, pp. 396–413, 2020.

[82] B.-J. Hsu, I. Shen, D. Eide, A. Chen, and R. Rogahn, "Microsoft academic graph," https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/.

[83] "DOI foundation," https://www.doi.org/.

[84] "Pubmed," https://pubmed.ncbi.nlm.nih.gov/.

[85] "DBLP," https://dblp.org/.

[86] "Pubmed central," https://www.ncbi.nlm.nih.gov/pmc/.

[87] "arXiv," https://arxiv.org/.

[88] "Acl anthology," https://aclanthology.org/.

[89] "Scidocs - the dataset evaluation suite for specter," https://github.com/allenai/scidocs.

[90] A. Singh, M. D'Arcy, A. Cohan, D. Downey, and S. Feldman, "Scirepeval: A multi-format benchmark for scientific document representations," https://github.com/allenai/scirepeval.

[91] ——, "Scirepeval: A multi-format benchmark for scientific document representations," *ArXiv*, vol. abs/2211.13308, 2022.

[92] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *Advances in neural information processing systems*, vol. 33, pp. 22 118–22 133, 2020.

[93] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec, "Ogb-lsc: A large-scale challenge for machine learning on graphs," *arXiv preprint arXiv:2103.09430*, 2021.

[94] J. Priem, H. A. Piwowar, and R. Orr, "Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts," *ArXiv*, vol. abs/2205.01833, 2022.

[95] "Social science research network," https://papers.ssrn.com/sol3/DisplayJournalBrowse.cfm.

[96] https://research.northeastern.edu/about/institutes-centers/.

[97] J. Priem, H. Piwowar, and R. Orr, "Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts," 2022. [Online]. Available: https://arxiv.org/abs/2205.01833

[98] K. Church, “Better together team github,” https://github.com/kwchurch/JSALT_Better_Together, 2023.

[99] https://jsalt2023.univ-lemans.fr/en/better-together-text-context.html, 2023.