Enriching Scientific Paper Embeddings with Citation Context

Kevin Henner

A thesis submitted
in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2019

Committee:

Gina-Anne Levow, Chair

Arman Cohan, Chair

Program Authorized to Offer Degree:
Linguistics

University of Washington

Abstract

Enriching Scientific Paper Embeddings with Citation Context

Kevin Henner

Co-Chairs of the Supervisory Committee:
Gina-Anne Levow
University of Washington

Arman Cohan
Allen Institute for Artificial Intelligence

Amid profusion of scientific literature, methods to organize and search available papers are quite valuable. Embedded representations of papers have potential to be used as input to a variety of tasks related to research paper search and recommendation. Such methods typically focus on document content, though some incorporate citation information. This citation information, however, is generally treated as fungible, with any citation given equal weight and identical meaning as any other. Recent advances in automated citation classification allow citations to be classified according how they are used in the citing document. I present a novel method for incorporating intent information into scientific paper embeddings through edge-weighting and concatenation of per-intent node2vec embeddings. Furthermore, I suggest that a hybrid approach, including both text and network data to generate embeddings can take advantage of both complementary and reinforcing information to provide a fuller embedded representation. I evaluate these embeddings on a set of classification and sequence modeling tasks. The results show a significant improvement in some, but not all cases, suggesting that while the incorporation of citation intent classification into scientific paper embeddings is promising, further work is needed to assess whether it can out-perform state-of-the-art alternatives and to further elucidate its contributions.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

Because his name doesn't appear on this project in an official capacity, I should first acknowledge the significant contributions of David Jurgens, without whose input, mentorship, and food-themed computer infrastructure this project have been impossible. I'm also incredibly grateful to Arman Cohan for his support and advice from conception to the end of this project, particularly his invaluable help in getting started with BERT in AllenNLP, and his thorough feedback on drafts. I own much to Gina-Anne Levow, who helped so much as I felt my way towards an appropriate topic, offered guidance throughout the thesis process, and provided excellent draft feedback. Much credit is also due to the other faculty and students in the University of Washington CLMS program and Linguistics Department, particularly, Emily Bender, Fei Xia, and Amandalynne Paullada.

DEDICATION

This work is dedicated to my parents, Jane and Dennis.

Chapter 1

INTRODUCTION

Beneath the ever-accelerating (Bornmann and Mutz, 2014) profusion of scientific literature, a researcher's challenge is as much in the navigation and triage of available work as in its digestion and synthesis. The dimensions of this task are as varied as those of the scientific process itself: a researcher seeks knowledge of a field's objects, however diverse or broadly conceived, but also its history, practitioners, trends, and methods. This search is shaped by sociological, political, and financial considerations–personal loyalties, pet theories, grant-worthy topics, druggable families of proteins–as well as dimensions of organization and accessibility–in what language is the paper written? How is it indexed and ranked by commonly used search engines? Does it require a subscription to access?

Embedding methods, which have shown great capacity for capturing the similarly nuanced semantic and syntactical dimensions of language, suggest a possible approach to this complexity. Rather than narrow our focus to some specific set of dimensions in the scientific literature that we suppose a researcher may find valuable, an effective embedding method may effectively encode a broad swath of this information in such a way that it can be elicited according to a variety of tasks and goals.

While part of such an embedding approach may focus on the text content of scientific papers, much valuable structure is present in the citation network. Citations reveal relationships among articles, among the topics those articles address, and among the authors and institutions involved in their research and publication. This network of relationships maps to a directed graph data structure wherein nodes and edges correspond, respectively, to a set of publications and citations.

As a researcher engages with a citation network, whether as an implicit cognitive model

alone or as supported and made explicit by research tools, she may use the relationships in this network to identify and evaluate works, authors, and topics relevant to her own research goals, and better understand the histories, trends, and divisions that characterize the collective literature of a field or subfield. However, the creation and interpretation involves far more than the simple indexical pointing from one text to another implied by a de-contextualized citation graph. The reasons an author might include a citation are diverse (Tahamtan and Bornmann, 2018), as are a reader's range of interpretations (Hicks and Potter, 1991).

If a large proportion of citations have a relatively low impact on a citing paper (Valenzuela et al. 2015), we hypothesize that these low impact citations add significant noise to a uniformly weighted citation graph. Though background citations may contribute information related to a document's coarser grained semantic position, this information may already be accounted for by higher order citation connections. Giving such citations equal weight, then, may reduce the local semantic discriminative potential of derived document embeddings.

Recent work on automated citation function classification (e.g. Hernández-Alvarez et al. 2017, Valenzuela et al. 2015, Abu-Jbara et al. 2013) raises the possibility of adapting existing bibliometric analysis of large corpora to account for different ways authors use citations in their work.

In the same sense that Firth's observation that "you shall know a word by the company it keeps" describes word embeddings, a publication may be represented as a function of its position in a citation graph. Works that cite and are cited by overlapping sets of publications have a more similar role in a body of scientific work, while those whose citations do not match play distinct roles. Furthermore, just as latent factors in embedded text representations capture similarity among words that do not directly co-occur (Landauer et al., 1998), citation-based document embeddings may capture important relationships among publications not connected through direct or triangular edge relationships in the underlying citation graph.

The vector document representations derived from a citation graph embedding process are suitable for a variety of information retrieval and summarization tasks. As such, they present

opportunities for extrinsic evaluation of citation intent classification through integration into these tasks.

1.1 Contributions

This work focuses on two key contributions.

First, the significant contributions of word representations to a variety of NLP related tasks show the versatility of dimensionally reduced vector representations. Vector representations of scientific papers may be used for a variety of tasks such as summarization, document recommendation and retrieval, citation prediction, and trend prediction. This work contributes a model that incorporates both textual and bibliometric data into the embedded paper representation, allowing sensitivity to both the positive content of a paper and its position in a broader discourse. Furthermore, by including citation intent data in the model's bibliometric component, I allow the model to consider *how* a paper is used in this discourse, rather than just *where* it is positioned in the network.

Second, this work contributes to the understanding of the information structure and semantics of citations in scientific communication. Bibliometric data have applications across the strata of the scientific process, from tools and heuristics to guide a literature search, to research management and evaluation, to the reflexive study of scientific processes themselves. In each of these areas, citation counts and their families of derivative indices are convenient quantitative proxies for key desiderata. Important questions around the appropriate interpretation of these data, however, remain unsettled. By demonstrating the contribution of citation intent to an embedding model, this work suggests that citation intent–and perhaps citation context more generally–should be considered in other uses of bibliometric data.

1.2 Chapter Outline

In Chapter 2, I discuss prior work and frame this project's precedents and contribution. I address bibliometrics, citation classification, research paper recommender systems, and scientific paper embedding methods.

In Chapter 3, I describe my methodology. I describe my approach to text embedding, the system I use for automated intent classification, my approach to node embeddings, and the method I use to create a hybrid embedding model from concatenated network node embeddings and text embeddings.

Chapter 4 includes a description of data, setup, and results from two classification tasks and a sequence modeling task.

Chapter 5 includes an error analysis and qualitative discussion of experiments in Chapter 4.

Chapter 6 is a discussion of conclusions and future work.

Chapter 2

RELATED WORK

Broadly, the background of this work splits into the bodies of literature related to bibliometrics and citation classification, research paper recommender systems, and document embedding methods. This distinction is not a clean one–some relevant works cross cut or exceed these categories–but should serve well to organize the following review.

I first review the history of bibliometrics to contextualize the origins and role of citation function and intent classification, then address the background and recent progress of automated citation classification systems.

Next, I briefly discuss recommendation systems in general, then address the specifics of scientific paper recommendation. I review the role of content-based and bibliometric recommendation methods, and address recent work in these areas.

Finally, I will discuss the background of text and graph embeddings and their applications in scientific paper recommendation systems.

2.1 Bibliometrics and citation classification

By including a citation, the author of a scholarly work marks an explicit association between a passage in his or her own text and the cited work. In aggregate, this form of explicit reference lets us construct a citation network, a kind of skeletal map of co-textual ties across the body of scientific discourse. Given the vast and ever-growing mass of literature, such structure offers an invaluable analytical and navigational resource.

Though the first statistical investigations of citation data date back to the beginning of the 20th century, the sixties, particularly with the creation and publication of the Science Citation Index (Garfield, 1964), mark a shift from an era of time-consuming and *ad-hoc*

collation of bibliographic data to larger scale and continually maintained indexing (Godin, 2006).

This post SCI era saw an increase in the use of bibliometric indicators in research evaluation and management. In the 1980's and 90's, a set of policies associated with New Public Management emphasized the importation economic rationalism from corporations to the public sector, including public research institutions and funding mechanisms (Gingras, 2016). Bibliometric indicators reduce complex and contextually embedded matters of scientific quality and productivity to numeric metrics legible to an economic idiom of rational maximization. This larger scale and institutional uptake add weight to questions about citations' role in scientific discourse.

Though the abstraction from context makes citations countable, the position and linguistic context of a citation anchor in the citing work contributes to its semantic value. This context can characterize the cited paper, frame it in relation to the citing work and its discourse structure, and position it in relation to other works or a broader conversation (Tahamtan and Bornmann, 2018).

In the half-century or so following the first publication of SCI, researchers developed a variety of annotations to describe and analyze the various function of citations in the citing work. The first such system (Lipetz, 1965) was a direct adaptation from that used in *Shephard's Citations*, an index of citations in court decisions, annotated to indicate whether a precedent was questioned, clarified, modified, or overruled. Garfield (1965) classifies citations by 15 types of authorial motivation such as "paying homage to pioneers", "providing background reading", and "substantiating claims". Since, a variety of annotation schemes and methods have been introduced, following a variety of thematic foci and annotation methods, and built on different collections of documents (e.g. Hodges 1972, Chubin and Moitra, 1975, Moravcsik and Murugesan 1975, Spiegel-Rosing, 1977, Small 1982, Swales 1990).

Schemes that require manual annotation, however, are difficult to maintain or expand to a larger scale. As Teufel et al. (2006) note, these annotation schemes often lacked rigorous cross-validation and independent annotation studies, and generally covered only a small

number of examples. A perennial theme of this work, then, is the idea that this annotation might be done, as Garfield wrote, by "machines or machine-like people" (Garfield, 1965).

Increase in available compute power and improvements in natural language processing methods have since made automated classification of citation function feasible. Some early steps in this direction include methods to identify key words and phrases in the citation context.

Garzone and Mercer (2000) introduce the first fully automated approach, using a hand-made set of grammatical and lexical rules to categorize citations according to a rubric of 35 categories. Teufel et al. (2006) introduce the first machine learning citation classification method based on a set of features derived from the citation context and position of the citation in the cited work.

Others such as Agarwal et al. (2010) and Abu-Jbara et al. (2013) implement classifiers using a variety of machine learning methods, selected features, citation categories, and targeted domains. Abu-Jbara et al. (2013) classify citations according to both purpose and polarity, suggesting that commonly used bibliometric measures such as the *H-Index* (Hirsch, 2005), *G-index* (Egghe, 2006), and *Journal Impact Factor* (Garfield, 1994) suffer from their uniform treatment of all citations.

In keeping with this theme that "not all citations are equal," Valenzuela et al. (2015) implement an automated method to assess the importance of each citation to the citing work. Their fine and coarse-grained importance labels are mapped directly from a citation type classification, as shown in Table 2.1. The classifier uses a variety of features, including those derived from the location of the cited paper in the text, the textual context of the citation, and bibliometric features of the cited paper (Valenzuela et al., 2015).

Hassan et al. (2017) explore the effectiveness of a variety of machine learning approaches to classify citation importance, treating use and extension function categories as important, while background and result comparison are non-important.

Zhu et al. (2015) similarly propose an SVM method to identify references with a high "academic influence" on the citing paper based on a set of features including citation location,

Citation Type	Fine-grained Label	Coarse Label
Related work	0	Incidental
Comparison	1	Incidental
Using the work	2	Important
Extending the work	3	Important

Table 2.1: Citation annotation labels in Valenzuela et al. (2015).

semantic similarity across the citing and cited papers, the cited paper's overall citation count, and the number of times the paper is cited within the citing paper. As a gold-standard, they surveyed paper authors for their own assessments of paper importance. Hernández-Alvarez et al. (2017) propose a multi-dimensional scheme to classify citations according to their function, polarity, aspects, and influence.

Jurgens et al. (2018) focus on authors' framing of citations within the citation context. They use a set of six classes, as shown in Table 2.2. They train a classifier on a manually annotated dataset, and use predicted categories to analyze dynamics of citation framing in the field of Natural Language Processing. Like the works mentioned above, their model incorporates a variety of textual, positional, and bibliometric features.

Cohan et al. (2019) present a neural method for citation intent classification. Their model uses concatenated contextualized ELMo (Peters et al., 2018) and non-contextualzed GloVe (Pennington et al., 2014) word embeddings as input, and incorporates bi-directional LSTM and attention hidden layers. As citation intent training data must be manually annotated, and is thus limited in size, the model, drawing on the multi-task approach described in Caruana (1997) incorporates auxiliary "scaffold" tasks based on easily extracted citation-worthiness and section data. This neural approach represents a significant break from the reliance of previous methods on feature engineering.

Cohan et al. (2019) also introduce an annotated SciCite corpus, drawn from 6,627 papers

Class (Distribution)	Description
Background (0.51)	P provides relevant information for this domain.
Uses (0.19)	Uses data, methods, etc. from P
Compares or Contrasts (0.18)	Expresses similarity/differences to P
Motivation (0.05)	P illustrates need for data, goals, methods, etc.
Extension (0.04)	Extends P's data, methods, etc.
Future (0.04)	P is a potential avenue for future work

Table 2.2: Citation class descriptions and distributions from Jurgens et al. (2018). P indicates the cited paper. Frequencies from Jurgens et al. (2018) are for citations in a manually annotated set of 52 papers drawn from the ARC (Bird et al., 2008).

in the computer science and biomedical domains, and including a total of 11,020 citation sentences annotated with a simplified three-class scheme (Table 2.3). The use of a multi-domain training corpus, simplified annotation scheme, and the reduced data pre-processing requirements of the neural method broaden the applicability of this approach.

Beltagy et al. (2019) improve further on these results with SciBERT, a BERT (Devlin et al., 2018) model pre-trained on scientific text. While the SciBERT model outperforms the scaffolded model presented in Cohan et al. (2019), at the time of writing, a scaffolded approach with the SciBERT model has not been attempted.

The current work applies this state-of-the-art citation classification method presented in Beltagy et al. (2019) to a scientific paper embedding system.

2.2 Research paper recommender systems

Paper recommender systems generally have two faces: a document indexing or representation system to encode the papers that may be returned, and a query or user modeling system used to make some selection of those papers. As this project is focused on document embeddings,

Class (Distribution)	Description
Background (0.58)	The citation states, mentions, or points to the background information giving more context about a problem, concept, approach, topic, or importance of the problem in the field
Method (0.29)	Making use of a method, tool, approach or dataset
Result (0.13)	Comparison of the paper's results/findings with the results/findings of other work

Table 2.3: Citation class descriptions and distributions in the SciCite corpus from Cohan et al. (2019).

I will restrict my discussion of the prior literature to this first aspect.

Bollacker et al. (1998) introduced the first research paper recommender system as part of the CiteSeer project. Recommendations are made based on a weighted combination of several paper similarity metrics, a TF-IDF bag-of-words model from paper text, string edit distance of document metadata, and a Common Citation Inverse Document Frequency (CC-IDF) analogous to text TF-IDF, but based on document citations rather than text tokens.

Beel et al. (2016) provide a review of research-paper recommender systems from 1998 to 2013, addressing 217 works published in that time span. This review found that only 16% of these papers used graph-based recommendations, while the majority (55%) use content-based methods. Of the reviewed papers using a content-based approach, most used plain words extracted from the paper text.

Along with features derived from paper content and a user's previous behavior, Bethard and Jurafsky (2010) use bibliometric features such as citation counts, author h-index, and citation context to train a model for paper recommendation. Zarrinkalam and Kahani (2012)

have developed a method to rate the similarity of documents based on available citation and metadata features, to allow for recommendations in contexts where full text is not available.

Work discussing the creation of a literature graph for Semantic Scholar (Ammar et al., 2018) notes in passing that these citation influence classifications have been used in the graph, but does not address whether or how these data are used in search or recommendation systems. Also associated with this project, Bhagavatula et al. (2018) describe a method to re-rank search results derived from a document embedding k-nearest neighbors model. This re-ranking system uses a neural network trained on text content features from a source and target paper to estimate the probability of a citation between the pair.

Kanakia et al. (2019) describe a hybrid recommender system used by Microsoft Academic. Their method uses a normalized linear combination of TF-IDF weighted *word2vec* (Mikolov et al., 2013) embeddings from the paper text to create a paper content embedding. An additional candidate set is derived from a co-citation model originally described by Small (1973). While paper candidates from both sets are combined in the model's output, the co-citation and content-based models are independent.

2.3 Scientific paper embedding methods

Scientific papers are characterized both by their position in a citation graph and their textual content. As such, I find relevant prior work related to both text embedding and graph embedding methods.

I hypothesize that the text content and citation network context of academic papers contains both complementary and reinforcing information salient to a scholar's research goals. While bibliometric features like co-citation and citation counts help a researcher efficiently identify key contributions to a field, text content may suggest connections otherwise overlooked by subfield communities. Furthermore, citations exist within a textual context that establishes their discourse role.

Before addressing the specifics of scientific paper embedding, I begin with a review of embedding methods in general. An embedding algorithm encodes input examples such as

words, sentences, images, or documents as real valued vectors that preserve information salient to some task or set of tasks. These embedded representations can then be used as input values for those tasks, or generalized to other related tasks.

2.3.1 Text embedding

In the NLP domain, word embedding models trace a lineage back to other forms of text representation, starting with bag of word models and re-weighting transformations like TF-IDF. By such models, a text can be represented as a real valued vector of dimensionality equal to the vocabulary size. This vector representation allows one to assess text similarity, for example, by simple vector operations such as dot product or cosine similarity.

Early methods such as Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003) reduce the dimensionality of the word occurrence matrix from a corpus, allowing a dense matrix of lower dimensionality to capture much of the word co-occurrence information in the original data. Furthermore, while these models may still be used to represent larger texts, they also provide per-word representations, which are useful in a wide variety NLP tasks.

As Landauer and Dumais (1997) note, the dimensionality reduction also has the effect of identifying latent dimensions in the data. This allows the model to capture the similarity of words that never co-occur in the same document.

However, as these models must operate on the whole matrix, they become prohibitively expensive on large datasets. Feedforward neural architectures, first seen in NNLM (Bengio et al., 2003) and popularized by word2vec (Mikolov et al., 2013) address this "curse of dimensionality" by learning per-word vector representations through the iterative neural network training process. The refinements introduced in Mikolov et al. (2013) allow a word embedding model to be trained on a very large data set and the learned word representations to be then transferred for use as word input representations in novel tasks.

BERT (Devlin et al., 2018) expands on a body of work that has effectively applied language model pre-training and transfer learning to a variety of natural language tasks.

Broadly, pre-training uses large datasets and significant compute resources to learn tasks designed to elicit generalizable intermediate representation of the input data. Transfer of these pre-trained models to other tasks may then improve performance and reduce training time and data needs.

In case of language model pre-training, for example, the goal is to learn generalizable word token representations. The semantic and syntactic information encoded in these representations then provides a linguistically rich foundation for modeling other tasks, relieving such models from the considerable work of learning these linguistic patterns *de novo*.

There are two established strategies for transfer learning: *feature-based* and *fine-tuning*. In the former, some vector representation is learned, which can then be mapped by an embedder to the input tokens. BERT is an example of the latter *fine-tuning* strategy: rather than extracting discrete vector representations, the whole trained model is transferred. Thus fine-tuning allows gradients from the new task's loss function to feed back through the pre-trained model.

While I refer the reader to the original paper (Devlin et al., 2018) for a full description of the BERT architecture, I will summarize the model's key features to help a reader understand the contribution of this model to our current methodology.

The BERT model consists of the following three key components, which I will summarize below: the *input representation*, the *transformer architecture*, and the *pre-training tasks*. We will then discuss the SciBERT pre-trained model and the specifics of our own implementation.

The first step in BERT input is a WordPiece tokenizer with a (Wu et al. 2016) vocabulary of 30,000 (Devlin et al. 2018). WordPiece tokenization uses a data-driven process to derive deterministic segmentations for any possible character sequence, thus keeping common words whole while avoiding OOV by segmenting less common words into word-piece tokens.

A [CLS] token is prepended to the input sequence. This token's value vector is used in classification tasks as an aggregate sequence representation. When multiple sentences are encoded, a [SEP] token separates them and represents the relationship between sequences in

sentence-pair classification tasks (Devlin et al. 2018).

In addition to these token sequence embeddings, the BERT input representation includes corresponding sequences of segment and position embeddings. The position embeddings encode an index position of the word-piece in the input sequence. As BERT uses a bidirectional transformer model that allows connections between all inputs in the sequence, these positional embeddings give the model sensitivity to relative and absolute word position. Segment embeddings encode a token's membership in sentence A or B when two sentences are encoded. This allows the model to learn distinct sub- and supra-sentential relationships among tokens. The final input to the model is the per-token sum of these three input layers.

The BERT model architecture consists of stacked multi-headed bidirectional Transformer encoder layers. The attention function learns a query, key, and value representation for each input token, calculated in each layer as:

$$\text{Attention}(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where d_k is the dimension of K (Rush 2018). Multi-heading maintains multiple Q, K, and V, matrices, allowing the model to jointly learn multiple representational subspaces and thus distinct relationships among tokens across those subspaces (Rush 2018).

During pre-training and refinement, the model learns both token representations and relationships among tokens. Including multiple layers allows this contextualization process to be repeated a number of times, and thus for the contextualized representation at each level to contribute context at the next.

BERT uses two pre-training tasks to help encode both word-to-word and sentence-to-sentence relationships. The masked language modeling (MLM) task allows the bi-directional multi-layer transformation architecture to learn relationships among words without "peeking ahead" as it would in a traditional word-sequence prediction task. To learn relationships between sentences, the model is trained to distinguish between pairs of sentences that are adjacent in the training corpus and randomly selected sentence pairs.

While BERT itself has been very effective for a variety of tasks, it relies only on data

encoded in the text input. As far as I am aware, BERT has not been used in a hybrid model along with node embedding data as a method of document embeddings or recommendation.

2.3.2 Node embeddings

A *graph* G consists of a tuple of nodes, edges, and adjacency matrix (V, E, A). I denote a given node as $v_i \in V$ and a given edge as $e_{ij} = (v_i, v_j) \in E$. The adjacency matrix consists of edge weights $w_{ij} = A_{ij}$ such that $w_{ij} > 0 \leftrightarrow e_{ij} \in E$. A node's first-order neighbors $N(v)$ are the set of all nodes connected by an edge: $N(v_i) = \{v_j \in V | e_{ij} \in E\}$. $N_S(v)$ to denotes a neighborhood near v sampled by method S.

Formally, a node embedding function is a mapping function $f : A \in \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times d}$ where $d \ll N$. The embedding for node v_i corresponds to matrix row A_i, and may also be denoted as $f(v_i)$. To be useful, an embedding must preserve some aspect of the original graph's topology. Exactly what information is preserved varies according to the specifics of the implementation. Furthermore, as is the case in word embedding methods, the embedding process may identify latent features. Thus in the citation network case, two papers that are never connected by a citation may still be nearby in the embedded space.

Incorporating network information into neural models presents a set of challenges distinct from the linear form of text or audio data or the two-dimensional raster representation common in computer vision applications (Wu et al., 2019b). As any given node in a network has the potential to be connected to any other, methods like convolutions, which rely on a fixed context size, cannot be applied to network data without significant modification.

In their survey on the topic Wu et al. (2019b) distinguish two broad approaches to deep learning with network data. *Graph neural network* approaches such as FastGCN (Chen et al., 2018) and SGC (Wu et al., 2019a) include the full adjacency data of the graph at training time.

Network embedding approaches represent graph vertices in a low dimensional vector space in a way that preserves information of the network's structure. These embedded representations can then be easily used as inputs to other machine learning algorithms.

Wu et al. (2019b) further subdivide network embedding approaches into *matrix factorization* and *random walk* methods. Matrix factorization methods such as BANE (Yang et al., 2018) and BNE (Shen et al., 2018) decompose a network's adjacency matrix into lower dimensional latent factors which may then be used as node representations. Random walk methods such DeepWalk (Perozzi et al., 2014), LINE (Tang and Qu, 2015), and Node2Vec (Grover and Leskovec, 2016) traverse the citation graph to generate a set of fixed length walks. These walks provide sets of linear node contexts more easily incorporated into an objective function for learning embeddings.

Attributed network embeddings such as BANE (Yang et al., 2018) and GraphSAGE (Hamilton et al., 2017) incorporate node attribute data into the embedding process, which allows the model to generate embeddings and perform link prediction on previously unknown data.

As my intent is to evaluate the contribution of citation intent classification in scientific paper, rather than directly optimize task performance, I selected node2vec as a strong compromise between performance, efficiency, and simplicity. (Grover and Leskovec, 2016) provide a reference implementation that works well at the scale of the ACL-ARC dataset, allowing me to evaluate a variety of variations and hyperparameter settings. To better frame my extension of the node2vec approach I review that approach in more detail.

The node2vec algorithm improves on prior random walk methods by introducing bias parameters that allow the walk to interpolate between depth-first and breadth-first strategies. Given current node $v = c_i$ and previous node $t = c_{i-1}$, a depth-first strategy will always prefer to visit some next node x such that $(t, x) \notin E$, while a breadth-first strategy will always prefer to visit a next node x such that $(t, x) \in E$. The breadth-first walk tends to produce sequences that remain in interconnected local clusters, while a depth-first walk tends to branch out into new regions of the graph.

Given p and q as parameters of bias function α, the un-normalized transition probability is $\pi_{vx} = \alpha_{pq}(t, x) * w_{vx}$, where t and x are the last-visited and next-visited nodes relative to

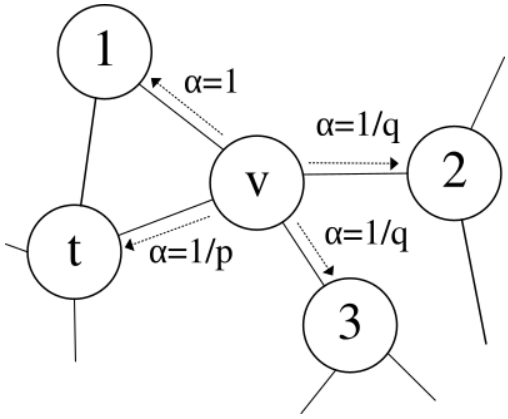


Figure 2.1: Illustration of the bias parameters for a random walk where t and v are the previous and current nodes, and α is the un-normalized transition probability corresponding to each edge.

v, and w_{vx} is the corresponding edge weight. The bias function α_{pq} is defined as follows:

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } t = x \\ 1 & \text{if } (t,x) \in E \\ \frac{1}{q} & \text{if } (t,x) \notin E \end{cases}$$

Grover and Leskovec (2016) describe q as the *in-out* parameter. As shown in 2.1, it directly adjusts the probabilities of visiting nodes with an edge from the previously visited node and those without such an edge. Values of $q > 1$ bias the walk in favor of a breath-first strategy, generating walks that tend to sample a smaller local area. Values of $q < 1$ favor a depth-first strategy, generating walks that tend to sample outward from a local area.

The p term acts as an *return* parameter. High values of $p > max(q, 1)$ discourage return to the last-visited node, thereby reducing redundant 2-hop sampling. The value of p also impacts the outward tendency of the walk in a manner similar to q, as returning to the last-visited node will encourage local exploration.

The algorithm generates a total of $r \times |V|$ walks, where r is the number of walks centered on each node $u \in V$. Each walk continues for l steps, where l is a fixed walk length.

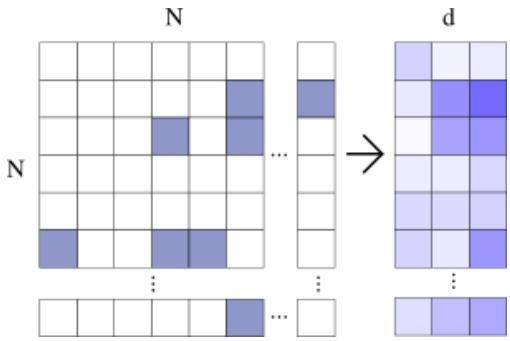


Figure 2.2: A node embedding maps an $N \times N$ adjacency matrix to an $N \times d$ node embedding matrix.

As each sampling step depends only on a single prior item, the process is second-order Markovian. As such, all transition probabilities can be pre-calculated, and the random walks can be fully parallelized. As graph data often includes a very large number of nodes and a larger number of edges, this scalability is an appealing property.

Grover and Leskovec (2016) show that adjusting this sampling strategy can shift emphasis between representation of a node's structural properties and its community membership. In other words, a node embedding can represent either the structural role of *how* a node is connected to those around it (e.g. as a hub or bridge), or the homophilic quality of *what* community of nodes it is connected to.

Once the set of random walks is defined node2vec uses a Skip-gram model (Mikolov et al., 2013) to sample node neighborhoods from these sequences. This sampling process works by sliding a fixed-width window across a sequence. For each index position i, it takes the item c_i as the target node, and the remaining nodes in the span c_{i-w}^{c+w} as that node's context. This corresponds to a given node $u \in V$ and a sampled neighborhood $N_S(u)$

The node2vec algorithm approaches the embedding learning problem as one of maximum likelihood optimization. The goal is to maximize the log probability of the observed network neighborhood $N_S(v)$ given node embedding $f(v)$.

$$\max_f \sum_{v \in V} logPr(N_S(v)|f(v)).$$

Assuming conditional independence and symmetry of a source node and neighborhood node in the feature space, this maximization problem may be simplified as follows:

$$\max_{f} \sum_{v \in V} \left(-\log Z_v + \sum_{n_i \in N_S(v)} f(n_i) \cdot f(v) \right),$$

where Z_v is a per-node partition function $\sum_{v' \in V} exp(f(v) \cdot f(v'))$.

This per-node partition function $Z_u = \sum_{v \in V} exp(f(v) \cdot f(u))$ is computationally expensive, making the stochastic gradient ascent objective infeasible for large networks. Negative sampling (NS) provides a feasible approximation. We replace the Z_v term with the estimation:

$$\sum_{j=1}^{k} log(\sigma(-f(v_j) \cdot f(u))),$$

where each negative sample v_j is drawn from distribution $P(v)$ proportional to d_v^{β} where d_v is the degree of node v and β is a hyper-parameter conventionally set to $\frac{3}{4}$ due to good empirical performance shown in (Mikolov et al., 2013).

While node2vec offers an effective and scalable approach to network embeddings, it does not offer a direct method for incorporating edge label data beyond simple edge weights. While I explore the efficacy of edge weighting, my model expands on the original node2vec approach by introducing a concatenation method for incorporating multiple edge dimensions into a network embedding.

Chapter 3
METHODOLOGY

3.1 Introduction

I hypothesize that the text content and citation network position of scientific papers contain both reinforcing and complementary signals. The citation graph reflects how a given paper is received into a broader structure of scientific discourse. I expect some correlation of this citation graph position to the text content: papers in a common network neighborhood are likely to address common topics. These topics, however, may cross-cut the subfields or specializations that shape the citation graph. As such, I propose a hybrid approach that uses a state-of-the-art language model to represent paper text content. By incorporating both text and citation network information, then, I intend to create an embedded representation that encodes relations among papers that cross-cut local citation network neighborhoods.

First, I introduce an extension to the node2vec (Grover and Leskovec, 2016) node embedding algorithm to incorporate citation intent as edge labels. Second, I describe the BERT text embedding method and the pre-trained SciBERT model. I use a concatenation of embeddings from these two approaches to create a combined embedded paper representation.

I use the publicly-available ACL Anthology Reference Corpus (ACL ARC) (Bird et al., 2008), a collection of full-text English language papers in the computational linguistics domain.

My approach to creating the initial paper embeddings consists of the following steps:

1. Create contextualized text embeddings from paper abstracts.

2. Extract a set of citation edges from the ACL ARC. From the citation context sentence, classify the citation intent of each edge according to a set of predefined categories.

3. Learn an embedded representation for each paper that encodes its position in the intent annotated citation graph.

4. Concatenate the text and node embeddings to create a synthetic embedded representation for each paper.

3.2 Paper text embedding

For text embeddings, I use SciBERT (Beltagy et al., 2019), a BERT model (Devlin et al., 2018) pre-trained on a corpus of English, full-text scientific papers. This model shows state-of-the art performance on both the ScienceCite (Cohan et al., 2019) and ACL-ARC (Jurgens et al., 2018) citation datasets. For tasks where the full-text of the paper is available, I encode the abstract. For tasks that involve papers external to the ACL-ARC corpus, I do not have access to paper abstracts and encode the paper title. Due to the considerable memory requirements of the BERT model architecture, I encode only the first 128 tokens of the abstract or title for each paper.

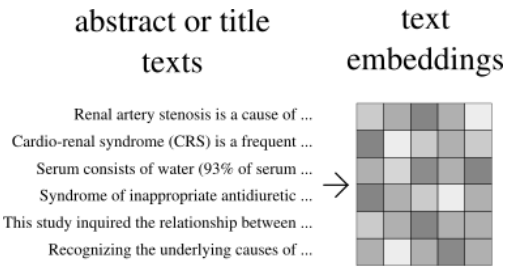


Figure 3.1: Text embeddings map input text sequences to real valued vector representations.

3.2.1 SciBERT

Because the ACL-ARC dataset uses specialized scientific language, I use the SciBERT pre-trained model (Beltagy et al., 2019). The SciBERT model is trained on a sample of 1.14M full text papers in the computer science and biomedical domains drawn from the Semantic

Scholar (Ammar et al., 2018). Beltagy et al. found that the SciBERT model outperforms a base BERT model for scientific text data on variety of tasks, including sequence tagging, dependency parsing, and text classification.

Following the best performing model variation found in Beltagy et al. (2019) I use an uncased SciBERT version with a SciVOCAB word-piece vocabulary derived from the pre-training corpus. I use the full SciBERT model, but allow refinement only of the top layer of the 12 stacked transformers in the model. This helps both to reduce the overall memory requirements of the experiments and to reduce overfitting. It is possible that with access to more resources and careful attention to tuning other model hyperparameters, allowing refinement of additional layers could improve further performance.

Depending on the task, I use either the paper abstract or title as input to the pre-trained SciBERT model. From the top layer of this model's transformer stack, I select the [CLS] token vector representation as the aggregate representation of the tokens in the embedded text, as suggested by Devlin et al. (2018). I then concatenate this vector to the same paper's node embedding to create a hybrid paper embedding representation. This concatenation allows downstream model layers to incorporate information from both the text and node embedding components.

3.3 Intent Classification

Following Cohan et al. (2019), I classify citation edges into BackgroundInformation, Method, and ResultComparison intent categories based on the sentence context of the citation anchor in the citing document. Cohan et al. (2019) found that additional categories of finer-grained classification schemes cover only a marginal percentage of citations.

As the pre-trained SciBERT achieves state-of-the-art results for this classification task I adopt this method as described in Beltagy et al. (2019).

For training the classification task, I use the SciCite corpus provided by Cohan et al. (2019). The corpus is drawn from a sample of 6,627 papers in the computer science and medical domains from the Semantic Scholar corpus. From these papers, Cohan et al. (2019)

use a combination of expert and crowd-sourced annotators to provide a total of 11,020 citation sentences annotated with intent classifications.

For a set of citations, the model takes the citation context sentence of each as input. During training, it refines the pre-trained SciBERT model. Once the model is trained on the original SciCite data, I use it to predicted citation intent annotations for each citation in the ACL ARC corpus.

3.4 Node embeddings

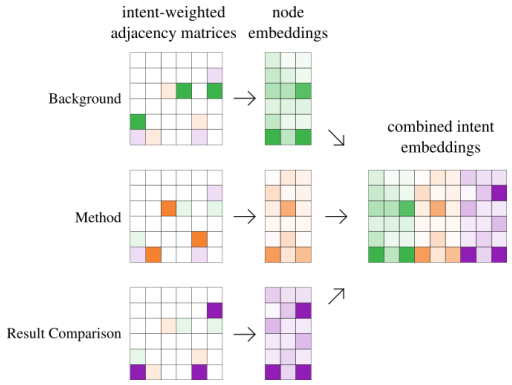


Figure 3.2: The concatenation approach concatenates three individually trained node2vec embedding matrices, each weighted to favor a different citation intent.

Formally, I add an intent dimension $k \in \{0, 1, 2\}$ to the set of edges E, and denote an edge in intent dimension k as $e_{ijk} = (v_i, v_j) \in E_k$. Note that as a citation to a given paper may occur multiple times in the citing paper with multiple intent classifications, this model allows an edge to exist between two given nodes in multiple intent dimensions.

Given this intent dimension, I implement two methods for incorporating intent into the random walks resulting node2vec embeddings: *weighting* and *concatenation*. In the weighting approach, I modify the base transition probability of each edge according to the associated intent. The weights for each intent are set as hyper-parameters. For example, the BACKGROUND category may be set to a low value to de-emphasize less impactful citations.

The concatenation approach extends the weighting approach by training a set of embeddings per-intent before concatenating these to yield a final combined embedding. For each of the three trained embeddings, an intent label is selected. Edges in the dimension corresponding to this dimension are weighted as 1, while the others are set to a lower weight whose exact value is set as a hyperparameter.

3.5 Hybrid embedding model

As SciBERT uses a refinement model while the node2vec embeddings are pre-trained, the model extracts the 768-dimensional vector value of the initial [CLS] token from the top layer of SciBERT's stacked transformer model. This is then concatenated with a trainable 384-dimensional node embedding vector.

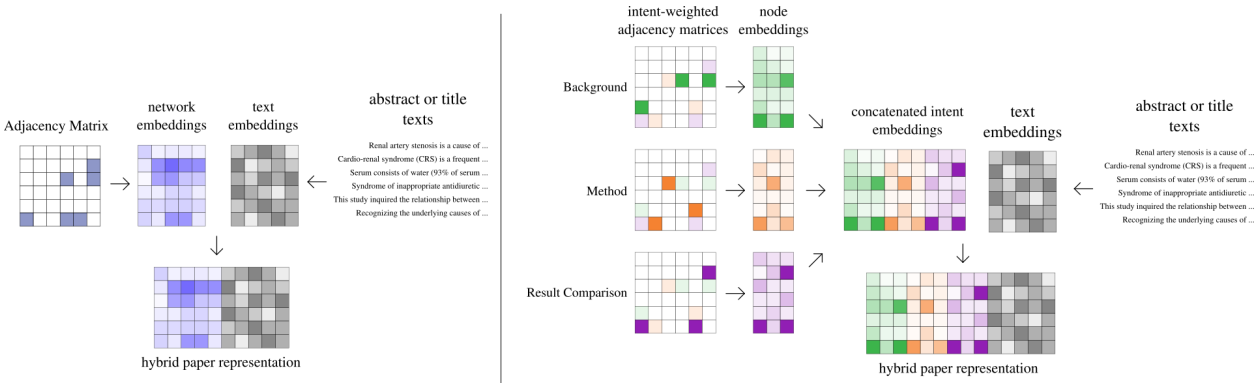


Figure 3.3: Overview of the weighted (left) and concatenated (right) hybrid paper representation models.

Chapter 4
EXPERIMENTS

As my goal is to develop a paper embedding method whose resulting representations are broadly adaptable, I evaluate the performance of the embeddings against three tasks. I select these tasks, in part, by the availability of data in the ACL-ARC corpus and the potential to scale or generalize to other similar corpora.

The experiments are as follows: *workshop category classification*, *last-author classification*, and *citation sequence modeling*.

I expect the workshop classification task to have a relatively narrow focus on topic similarity. Given a reasonable expectation that topics correlate well with abstract text, I expect text embedding models to make a significant contribution to this task. I expect a paper's last-author to correlate with a broader selection of topics, and for citation network embeddings to play a more significant role. The citation sequence modeling task may be significantly shaped both by topic, as the citing paper's author cites a series of papers in a section or passage with the same focus, and by citation network as an author engages with conversations in the literature or leverages these citation networks to discover related work.

4.1 Data

4.1.1 ACL ARC Corpus

I run my experiments on data from the ACL Anthology Reference Corpus (ACL ARC) (Bird et al., 2008), a collection of publicly available full-text English language papers in the computational linguistics domain. The collection includes a total of 18,288 articles from the year 2015 and earlier. I use a version of this corpus with citations identified by ParsCit (Councill et al., 2008) and pre-processed to a JSON format with canonicalized paper ids and

titles by Jurgens et al. (2018). Documents in the corpus include citations to a total of 99,577 external documents.

Note that as the original OCR conversion of the documents, the ParsCit process, and title text-matching in the canonicalization process are all fallible, the corpus is imperfect. Several papers, for example, appear multiple times under slight variations in title. As this represents, more-or-less, the state of "real-world" data in this domain, I do not believe that these issues have a significant impact on the results.

4.1.2 Sequence data preprocessing

As citation sequences include a large proportion of documents external to the corpus, I include these documents in the model (in contrast the the classification experiments). Because I lack full abstracts for these external documents, the SciBERT text representation in this experiment is based on paper title, rather than abstract.

Due to the large memory requirements of the SciBERT model, I split sequences into segments consisting of no more than 15 items. To represent the true beginning of a paper, I use a designated start token. This token's network embedding representation is randomly initialized, and its SciBERT text representation input consists of a single [unused0] token, which was reserved in the initial SciBERT training process and is thus also randomly initialized.

This dataset consists of $35,699$ sequences, which I split the data into training, validation, and test sets consisting of 80%, 10%, and 10% of the data, respectively.

4.2 Baseline comparisons

To evaluate the performance of the neural architecture without pre-training, I include a random initialization baseline with architecture and vector input sizes matching those of the other models.

As a non-neural baseline, for the last-author and workshop classification tasks, I use a TF-IDF weighted bag-of-words vector representation with a support vector machine classifier,

as implemented in the ScikitLearn package.

I compare the sequence model against an N-gram model to estimate probabilities of each paper-id token t in a session sequence conditional on the prior context. Using the Markov assumption, the N-gram model represents this prior context as $N-1$ prior tokens t_{i-N+1}^{i-1}. Thus I estimate $P(t_i|context(t_i)) \approx P(t_i|t_{i-N+1}^{i-1})$. The probability of a sequence, then, is the product of individual token probabilities: $P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-N+1}^{i-1})$ (Jurafsky and Martin, 2009). In the simplest form of an N-gram model, per-token N-gram probabilities are estimated as normalized counts from the training dataset: $P(t_i|t_{i-N+1}^{i-1}) = \frac{C(t_{i-N+1}^{i})}{C(t_{i-N+1}^{i-1})}$.

As a non-neural sequence model baseline, I use an interpolated trigram Kneser-Ney model (Kneser and Ney, 1993). Goodman (2001) found that this approach showed the second best performance among common trigram models. Though Goodman found that a modified Kneser-Ney model showed a small improvement, we follow the suggestion that the un-modified algorithm's smaller number of parameters makes it a more practical baseline.

Furthermore, while caching, class-based, and clustering variants of N-gram models may outperform the interpolated Kneser-Ney model (Goodman, 2001), these models rely on a linguistic context that cannot be trivially adapted to the present paper sequence model.

To implement the trigram model baseline, I use the SRILM toolkit (Stolcke, 2002), which provides an implementation of the Kneser-Ney model described above. As a further baseline, I also implement a unigram model, which simply orders papers by frequency across the corpus irrespective of prior context.

4.2.1 Model variations

I compare the un-modified and intent-annotated versions of node2vec embeddings with and without the SciBERT abstract embeddings, as well as the SciBERT model alone.

For all node2vec models, I use the following parameters: walk-length: 20, walks-per-source: 10, context-size: 10, p: 0.3, q: 0.7, directed: false. I used a limited grid-search to select these hyperparameters based on validation results for unweighted embeddings. This search included values of walk-length in $[10, 20, 40]$, and values of p and q in $[0.2, 0.3, 0.4, 0.5, 0.7]$.

I did not explore variation in the walks-per-source or context-size parameters, leaving these at default values suggested by Grover and Leskovec (2016).

I evaluate 5 variations of the node2vec network embeddings. The CONCAT prefix denotes a concatenated 384 dimension combination of a distinct 128 dimensional embedding calculated per-intent category. The ALL prefix denotes a 384 dimensional 'vanilla' node2vec embedding. I show the details of each weighting strategy in 4.1.

Model	description	BACKGROUND	METHOD	RESULT
ALL-U	uniform weights	1.0	1.0	1.0
ALL-B	background penalty	0.2	1.0	1.0
ALL-P	proportionate	~ 0.45	~ 0.58	~ 0.97
CONCAT-F	fixed weights for off-intent	$1.0; 0.5$	$1.0; 0.5$	$1.0; 0.5$
CONCAT-P	proportionate weights for off-intent	$1.0; \sim 0.45$	$1.0; \sim 0.58$	$1.0; \sim 0.97$

Table 4.1: Weighting strategy descriptions. For concatenated models, $1.0; 0.5$ denotes a weight of 1.0 for edges matching target intent category and 0.5 for off-intent edges.

4.3 Classification tasks

4.3.1 Experiment setup

Depending on the experimental condition, the paper representation layer consists of a node embedding, a text embedding, or a concatenation of both. A dropout of 0.3 is applied to this layer to reduce overfitting. This layer is passed forward to a 100 dimensional fully-connected layer with ReLU activation and a dropout of 0.4, then a softmax layer with dimensionality equal to the number of classes. I use a learning rate of $2e-5$. To reduce overfitting and reduce overall training time, I use early stopping with a patience of 15. I use a batch size of 16. Total epochs are limited to 150, however, due to early stopping, no model reached

this limit. I use the BERT Adam optimizer, a variant of the Adam stochastic optimization algorithm adapted to the BERT model.

4.3.2 Evaluation

To evaluate the performance of each model for the paper classification tasks, I compare macro-averaged precision, recall, and F1 scores as well as prediction accuracy. I run each model with 5 random seeds, and report evaluation results for the model with median test accuracy. To assess the significance of differences in models, I use McNemar's test on the set of predictions for the median accuracy result of each of the two models being compared.

4.3.3 Workshop classification

A useful paper embedding method should capture information related to document similarity and topicality. To assess this, I create a supervised classification task from ACL ARC conference workshop data. The ACL ARC dataset includes papers published in conference workshop sessions. To reduce data sparsity, I collapse sets of 1 or more workshop into thematically similar categories. I use 10 most frequent categories for the classification experiment. Table 4.2 shows these workshop categories and a selection of the associated titles for each.

This yields a total of $3,229$ instances, which I separated into 80% training data and 10% each validation and test data.

4.3.4 Results

As shown in Table 4.3, the SciBERT model performs well on this task, both with and without the inclusion of network embeddings. There is significant variation among the set of models with GloVe text embeddings, showing a significant contribution by the network embeddings. Notably, the GloVe n2v concat-p model (with proportional concatenated network embeddings) achieves an accuracy within the range of the SciBERT models.

4.4 Last author prediction

If workshop categories represent a relatively narrow topic focus, grouping papers by last author reflects a broader and more dynamic sample of that author's interests, institutional roles, and collaborative connections. If a paper embedding can capture this information, it demonstrates a capacity to encode this broader and more subtle pattern. While single author data is very sparse in the dataset, the last author on a paper is often in a senior or supervising position.

This yields a total of 538 instances, which I separated into 80% training data and 10% each validation and test data. The selected authors and corresponding counts are shown in Table 4.4. (Note that the set of author names is manually mapped to single categories from a set of variations, and it is very likely that this mapping is incomplete–this experiment is intended to test the performance of the embeddings in distinguishing among papers on which these authors' names appear, not to represent any author's productivity or importance.)

4.4.1 Results

Despite the attempt to reduce data sparsity by grouping by last author, this dataset was too small to provide a useful comparison among most of the models. While a pre-training approach can often yield good results even with relatively small training sets, it is apparent that, in this case, a larger training set is necessary. Furthermore, the small number of instances in the training and validation were problematic. Because classification of a few instances in small test and validation makes an outsize difference in metric outcomes, there was a high variance both in validation results during training and in the final test results across random seeds. While I expect that this task inherently more challenging than the workshop classification task, it isn't possible to disentangle this challenge from the problems of data sparsity.

4.5 Sequence model

An author's selection of citations and placement in the citing paper reflect a complex process of research and composition. A model's capacity to predict this sequence, then, may be improved both by recognizing patterns of similarity, but also patterns of discourse. That is, while I might expect semantically related documents to occur together, I should also see a reflection of a paper's discursive structure in the sequencing of papers it cites.

For this sequence modeling experiment, I extract sequences of citations as they occur in the text of the citing documents. For example, if the first three citations in a paper are *Scientific Paper Summarization Using Citation Summary Networks, Citation Summarization Through Keyphrase Extraction,* and *Multiple Alignment of Citation Sentences with Conditional Random Fields and Posterior Decoding*, the beginning of the corresponding sequence will consist of the ACL IDs for those papers prepended by a start token: [*<s>*, *C08–1087, C10–1101, D07–1089, ...*]

4.5.1 Evaluation

Perplexity

Perplexity is a common intrinsic evaluation metric for sequence modeling tasks. Perplexity measures the normalized inverse probability of all observations in a test dataset:

$$PP(T) = P(t_1 t_2 ... t_n)^{-\frac{1}{n}}$$

Because of this inverse component, lower values of perplexity correspond to a higher probability of the test data and imply a better model.

To interpret perplexity, it is useful to understand its relationship to cross-entropy. Mathematically, perplexity is an exponentiation of the cross-entropy estimated for the model on a sequence T:

$$PP(T) = 2^{H(T)},$$

where language L is a stationary ergodic process that produces a sequence of tokens with distribution p, the cross-entropy of model m on distribution p is a limit as the length of sequence $(t_1 t_2 ... t_n) = T \in L$ approaches infinity:

$$H(p, m) = \lim_{n \to \infty} -\frac{1}{n} \log m(t_1 t_2 ... t_n),$$

As our evaluation data is finite, this limit may be approximated for any given sufficiently long sequence T as $H(T) = -\frac{1}{n} log P(T)$ where $P(T)$ is the probability of T according to the model m.

The key observation, here, is that $H(L)$, the actual entropy of the language, is a lower bound on $H(p, m)$ for any model. As a model better approximates the distribution of a language itself, cross-entropy decreases. This means that even without knowing the actual entropy of a language, I can expect the model with the lowest cross entropy to be the best approximation.

While cross-entropy measures an average of bits per token to encode a sequence according to the given model, the exponentiation of cross-entropy to perplexity converts this to a weighted expected branching factor. In other words, a model that gives a uniform distribution over 10 possible tokens would have a perplexity of 10. As the model moves away from a uniform distribution, the branching factor is weighted according to this distribution, reducing the perplexity of the model.

Recall@N

While perplexity works well to assess relative performance of metrics, it can still be difficult to interpret concretely. To better assess how well the model is able to make appropriate predictions at each sequence position, I present a RECALL@N metric. For a given value N, RECALL@N represents average number of times the true paper-id token occurs within the top-N predicted tokens for a given sequence position.

4.5.2 Results

As shown in Table 4.6, the SciBERT model without network embeddings performs surprisingly poorly on this task–the Kneser-Ney trigram baseline as well as all variations of network embedding models outperform this model. Similarly, the hybrid models with both the SciBERT and network embedding components perform consistently worse than the corresponding models with only network embeddings, showing that the inclusion of the SciBERT text embedding model acts as a net drag on perplexity performance in this task. Note also that the SciBERT embeddings in this task were derived from document titles, rather than abstracts, which may contribute to the poorer overall performance.

Among the network embedding models with and without the SciBERT text embedding, there is a similar progression of improvement across the weighting variants, with the two concatenated models showing the best performance in both cases.

Plotting the Recall@N metrics across the models, however, shows a contrasting result. Here, the SciBERT n2v concat-p and SciBERT n2v concat-f models have the highest recall performance for all values of N.

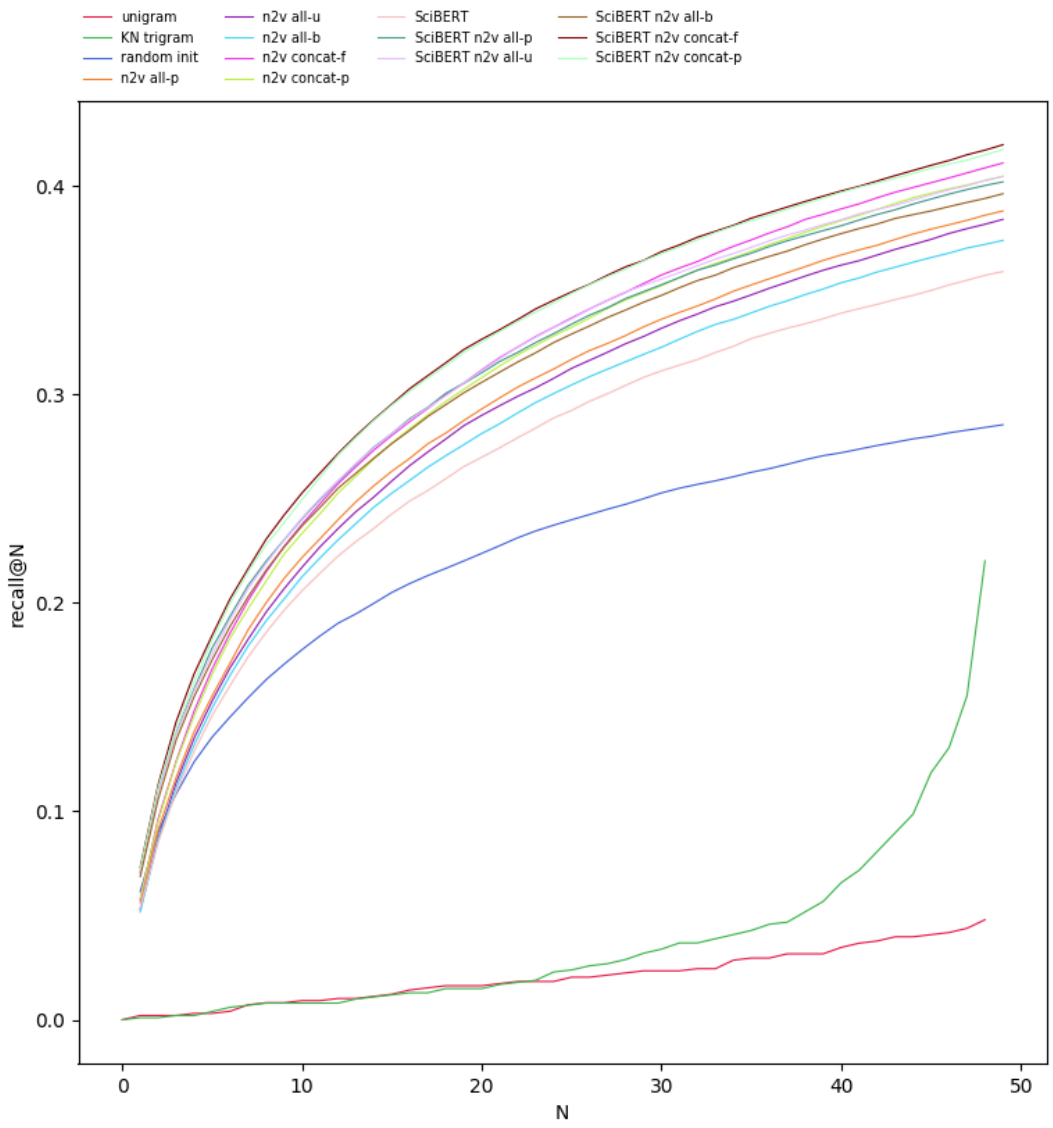


Figure 4.1: Recall@N plot for values of N from 0 to 50.

Category (count)	Example Workshop Titles
machine_translation (612)	Humans and Computer-assisted Translation Machine Translation Machine Translation Statistical Machine Translation and MetricsMATR Syntax, Semantics and Structure in Statistical Translation
dialogue (503)	Discourse and Dialogue Proceedings of the SIGDIAL Conference Companionable Dialogue Systems Evaluation Methodologies for Language and Dialogue Systems
natural_language_learning (398)	CoNLL Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop Proceedings of the Fifteenth Conference on Computational Natural Language Learning Cognitive Aspects of Computational Language Learning
semantics (395)	International Conference on Computational Semantics (IWCS) the Evaluation of Systems for the Semantic Analysis of Text Semantics in Text Processing. STEP Conference Proceedings Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)
nlg (332)	NLG Proceedings of the 6th International Natural Language Generation Conference Natural Language Generation (ENLG-05) Monolingual Text-To-Text Generation
bio_nlp (288)	Proceedings of BioNLP Workshop Biomedical Natural Language Processing Natural Language Processing in Biomedicine Natural Language Processing in the Biomedical Domain
annotation (210)	Linguistic Annotation Workshop Discourse Annotation Frontiers in Corpus Annotations II: Pie in the Sky Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL
chinese_language (200)	Chinese Language Processing
educational_applications (177)	Innovative Use of NLP for Building Educational Applications
multiword_expressions (114)	Multiword Expressions

Table 4.2: Workshop classification categories with the count of included papers and selected example workshop titles.

Model	P	R	F1	accuracy
RANDOM INIT	2.0	10.0	3.3	19.6
TF-IDF SVM	*63.6*	*63.5*	*62.9*	*73.3*
N2V ALL-P	*78.1*	72.6	*73.8*	77.3
N2V ALL-U	74.0	71.9	72.1	76.1
N2V ALL-B	70.7	70.8	70.2	73.6
N2V CONCAT-F	75.1	70.4	71.3	75.5
N2V CONCAT-P	72.9	*74.2*	73.4	*78.0*
GLOVE	66.3	58.9	60.2	70.8
GLOVE N2V ALL-P	76.9	78.0	77.4	79.8
GLOVE N2V ALL-U	77.2	78.7	77.5	79.8
GLOVE N2V ALL-B	77.0	77.2	76.8	79.8
GLOVE N2V CONCAT-F	74.4	74.9	74.4	78.9
GLOVE N2V CONCAT-P	*79.2*	*80.3*	*79.5*	*82.6*
SCIBERT	78.1	78.5	77.1	82.6
SCIBERT N2V ALL-P	79.9	**82.3**	**80.6**	82.9
SCIBERT N2V ALL-U	79.9	81.9	80.4	82.9
SCIBERT N2V ALL-B	**80.6**	80.6	80.0	82.6
SCIBERT N2V CONCAT-F	80.3	81.7	80.5	**83.2**
SCIBERT N2V CONCAT-P	76.7	78.3	76.9	82.3

Table 4.3: Results for classification of 10 grouped ACL workshop classes. The best result overall for each metric is rendered in bold, while the best result within each category is rendered in italics.

Category (count)
Christopher Manning (84)
Jun'ichi Tsujii (82)
Dan Klein (74)
Eduard Hovy (67)
Yuji Matsumoto (64)
Dan Roth (61)
Herman Ney (55)
Mirella Lapata (51)

Table 4.4: Last author categories with associated paper counts.

Model	P	R	F1	accuracy
RANDOM INITIALIZATION	3.3	10.0	4.9	13.8
TF-IDF SVM	*55.2*	*43.8*	*40.0*	*43.8*
N2V ALL-P	52.3	58.8	53.6	53.8
N2V ALL-U	59.1	51.9	47.8	50.0
N2V ALL-B	9.9	23.7	13.7	25.0
N2V CONCAT-F	**68.5**	*64.4*	*59.7*	**60.0**
N2V CONCAT-P	59.2	60.4	55.9	56.2
GLOVE	2.9	14.1	4.8	11.2
GLOVE N2V ALL-P	43.8	53.8	46.9	50.0
GLOVE N2V ALL-U	*64.8*	**66.5**	**61.2**	**60.0**
GLOVE N2V ALL-B	16.7	25.8	16.8	27.5
GLOVE N2V CONCAT-F	54.1	62.6	56.2	57.5
GLOVE N2V CONCAT-P	52.6	59.0	52.7	53.8
BERT	51.8	50.8	48.2	47.5
BERT N2V ALL-P	*55.1*	*53.3*	*51.0*	*50.0*
BERT N2V ALL-U	50.7	52.6	49.2	48.8
BERT N2V ALL-B	50.7	52.3	50.6	*50.0*
BERT N2V CONCAT-F	48.4	50.4	47.6	46.2
BERT N2V CONCAT-P	47.7	49.3	47.3	47.5

Table 4.5: Results for classification of 8 last author categories. The best result overall for each metric is rendered in bold, while the best result within each category is rendered in italics.

MODEL	PERPLEXITY
UNIGRAM	21807.1
KN TRIGRAM	*6373.5*
RANDOM INIT	12407.7
N2V ALL-B	5028.2
N2V ALL-U	4443.1
N2V ALL-P	4257.9
N2V CONCAT-P	3750.9
N2V CONCAT-F	**3604.1**
SCIBERT	6685.9
SCIBERT N2V ALL-B	4698.2
SCIBERT N2V ALL-U	4520.4
SCIBERT N2V ALL-P	4477.4
SCIBERT N2V CONCAT-P	4097.3
SCIBERT N2V CONCAT-F	*4084.2*

Table 4.6: Perplexity results for citation sequence modeling. Lower perplexity scores indicate better model performance. The best overall score is rendered in boldface, while best scores per section are italicised.

Chapter 5

DISCUSSION

Across the experiments, the results suggest that intent information contributes some information to paper embeddings. However, given the differences across the experiments and the across-the-board performance of the SciBERT model in the workshop classification task, further investigation will be necessary to fully understand the quality and impact of this contribution. Furthermore, while we see a similar pattern of stratification in the node2vec variations in the SciBERT and no SciBERT cases, the poor performance of the SciBERT model on test data for the sequence modeling task and very high performance on the training data suggests over-fitting. The baseline SciBERT model achieved a perplexity of 129.5 on the training data, for example.

The experiments show the clearest pattern in the node2vec alone and GloVe + node2vec cases for the workshop classification task. However, in the node2vec only case, McNemar's test shows a significant difference ($\alpha = 0.05$) only between the concat-p and all-b ($p = 0.035$), the strongest and weakest models in this category in terms of accuracy. McNemar's test does not show a significant difference between concat-p model and the uniformly weighted all-u model ($p = 0.345$). In the GloVe category, the concat-p model also shows the best accuracy, and McNemar's test shows a significant difference ($\alpha = 0.05$) between this model and all others in the category with the exception of the all-b model. The p values are shown in table 5.1.

Given the overall results, we cannot reject explanations for the performance of the concatenation models unrelated to the citation intent hypothesis. For example, it is possible that the redundancy introduced by the three concatenated node embeddings makes some contribution, rather than the differences among the three models. Of the two concatenation

MODEL	P-VALUE
GLOVE	< 0.001
GLOVE N2V ALL–U	0.014
GLOVE N2V ALL–P	0.008
GLOVE N2V ALL–B	0.091
GLOVE N2V CONCAT–F	0.005

Table 5.1: McNemar's test p values for the GLOVE N2V CONCAT-F model against other GLOVE models.

models, the one that makes the less drastic weighting difference among the categories seems to perform better.

My second question is whether a hybrid approach incorporating both network embedding and text embeddings can outperform either approach alone. While the overall strong performance of the SCIBERT model shows some evidence that a hybrid approach is not necessary for a strong result, the results for the GLOVE model clearly show a strong benefit of a hybrid approach, at least for this weaker text embedding model. In fact, McNemar's test did not show a significant improvement of any SCIBERT model over the best-performing hybrid GLOVE model. It may be that the SCIBERT model is already approaching the limits of accuracy inherent to the dataset, leaving no room for potential contributions from the network embeddings. It could also be the case that, though there is room for improvement in the dataset, the SCIBERT model already adequately captures whatever information is present in the network embedding model. Finally, it is possible that the model architecture is such that the model doesn't learn the include the network embedding contributions. As the SCIBERT models tend to converge more quickly than the network embedding and/or GLOVE models, it may be that the network embedding component's weights are reduced early in favor of the faster-learning *BERT* component.

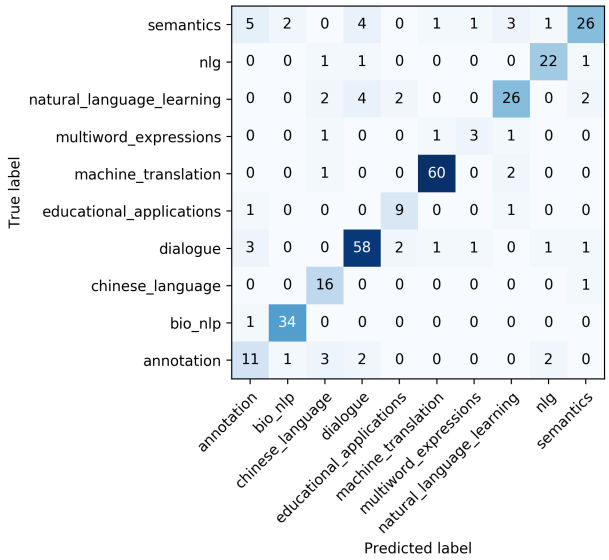


Figure 5.1: Confusion matrix for the SciBERT n2v concat-p model.

5.1 Error Analysis

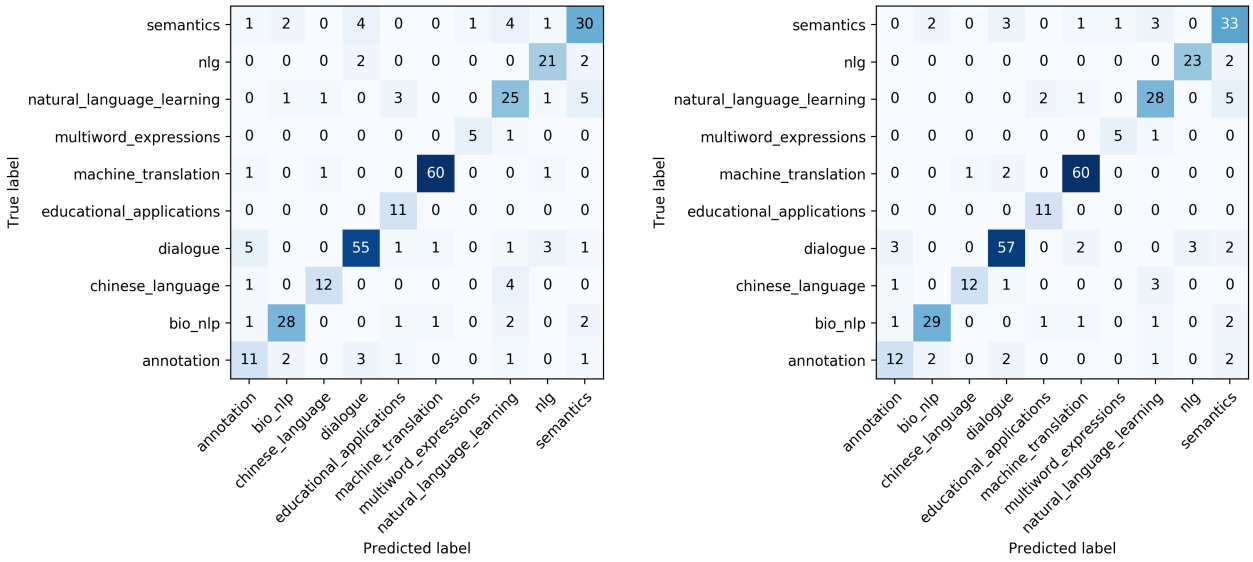
The confusion matrix for SciBERT n2v concat-p model (figure 5.1) shows the highest error rate for the *semantics* category, with the most common mis-classifications assigning these instances to the *annotation* and *dialogue* categories.

An examination of these examples (See table 5.2) shows that, subjectively, these mis-classified examples are at least as appropriate to the predicted category as the true category.

Comparing the confusion matrices between the GloVe n2v all-u and GloVe n2v concat-p models, there are fairly diffuse improvements across the categories. For example, several mis-classified instances in the dialogue category are corrected.

5.2 Qualitative Analysis

While quantitative metrics offer an essential basis for the comparison of models, the ultimate purpose of a paper embedding is for applications with significant subjective dimensions. If my



(a) GloVe n2v all-u model

(b) GloV n2v concat-p model

Figure 5.2: Comparison of confusion matrices for the GloVe n2v all-u model and GloVe n2v concat-p model.

predicted category	ACL ID	title
dialogue	W11-0101	The Semantics of Dialogue Acts
dialogue	W09-0506	Predicting Concept Types in User Corrections in Dialog
dialogue	W11-0119	The Exploitation of Spatial Information in Narrative Discourse
dialogue	W14-1402	System with Generalized Quantifiers on Dependent Types for Anaphora
annotation	W09-3706	Semantic annotations as complimentary to underspecified semantic representations
annotation	W09-0504	Semantic Representation of Non-Sentential Utterances in Dialog
annotation	W09-3716	GLML: Annotating Argument Selection and Coercion
annotation	W09-3723	Towards an Analysis of Opinions in News Editorials: How positive was the year? (project abstract)
annotation	W09-2406	Large-scale Semantic Networks: Annotation and Evaluation

Table 5.2: Mis-classification errors for the *semantics* workshop category.

intention is to create an embedding that broadly captures information that would be useful in a recommender system, search engine, or related application, qualitative investigation of the results are a necessary supplement to any quantitative result.

As shown in figure 5.3, plotting a T-SNE (van der Maaten and Hinton, 2008) dimensionality reduction from the uniformly weighted ALL-U node2vec embeddings and the concatenated proportional weight CONCAT-P embeddings shows clear clustering, even when reduced to two dimensions. Given the stochastic quality of the clusterings from different embeddings, interpreting these results has a Rorschach ink-blot quality. However, the CONCAT-P model does appear to show tighter grouping, particularly among some of peripheral sub-clusters in each

workshop category.

While the quantitative results from the sequence modeling task show significant differences among the models, the usefulness of sequence predictions depends on the absolute accuracy. As the model's recall is calculated based on a single "true" example in a linear sequence, this represents something like a minimum recall when we consider more broadly whether a predicted example is appropriate, whether or not it happens to be the one actually occurring in the given sequence. As such, some qualitative analysis of the results are needed to evaluate the degree to which the model can predict relevant citation sequences.

Unfortunately it seems that the papers predicted by the model are not very coherent. For example, starting with the sequence [<*s*>, *C08–1087, C10–1101*], which corresponds to papers with titles *scientific paper summarization using citation summary networks*, and *citation summarization through keyphrase extraction*, the true next item is *D07–1089*, titled *multiple alignment of citation sentences with conditional random fields and posterior decoding*. The titles of the top 5 predicted next items are as follows:

1. *an entitymention model for coreference resolution with inductive logic programming*
2. *an algorithm for finding noun phrase correspondences in bilingual corpora*
3. *chinese sentence segmentation as comma classification*
4. *a course in phonetics*
5. *ontonotes a unified relational semantic representation*

Given a recall@5 metric just under 20%, it is reasonable that this sample wouldn't return the correct result, but somewhat surprising that the predictions seem so poor. It may be that a relatively small set of sub-sequences or patterns are driving this recall metric, making the practical performance worse than the metric would suggest. In other words, the model may be performing well on some subset of sequences, while exhibiting poor generalizability to other sequences.

(a) N2V ALL-U model

(b) N2V CONCAT-P model

Figure 5.3: T-SNE comparison of the uniformly weighted ALL-U node2vec model (left two columns) and the concatenated proportional weight CONCAT-P model (right two columns).

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This work contributes a model that incorporates both text data and bibliometric network embeddings into embedded paper representations. Further, by including citation intent data in the network embedding component, I introduce information related to a paper's role in the citation network. The workshop classification task experiment showed a clear contribution of network embeddings when used in combination with the GloVe embeddings, and within the hybrid GloVe embedding models, there was a clear difference in performance based on citation intent weights, with the proportional concatenation model showing the best classification accuracy.

Similarly, in the case of the citation sequence model, there was a clear difference in performance according to the citation intent weighting strategies, both in combination with the SciBERT text embeddings and alone, with the fixed-weight concatenation model showing the best performance.

However, there remains some ambiguity in the results, which suggests that further work is necessary to elucidate the contributions of citation intent classification to scientific paper embedding methods.

6.2 Future Work

As the method of incorporation through concatenation of individually trained embeddings weighted to favor each intent category is quite simplistic, it seems that investigating further methods in this vein may be rewarding. The strong performance of the baseline SciBERT model suggests that the transformer architecture works well for the tasks. As such, a model

that better integrates this architecture with the initial node vector embedder trainer may be promising.

Given the near-uniform strong performance of the SciBERT model and plausibility of mis-classifications show in the error analysis, this model may be at or approaching the limit of classification accuracy for the workshop classification task. This suggests that a more challenging classification dataset may be needed to demonstrate an improved paper representation approach when the *SciBERT* model is used.

While I chose the node2vec approach for its simplicity, the large number of hyperparameters and variations led to a profusion of models. While neural graph model might be somewhat more complex to implement, and would certainly include hyperparameters of its own, moving these hyperparameters from an external node2vec training step into a single neural architecture would make for a simpler process and perhaps better search of the hyperparameter space.

The sequence modeling task may be easily adapted to other kinds of sequence data, such as trace data derived from user interactions with academic search engines. While such data would take the same form as the citation sequence data, it would represent a different stage in the research process. While sequencing of citations within papers may be more reflective of discursive structure within the citing paper, I expect user search behavior to be more directly applicable to modeling a research process.

Using a larger corpus, such as the PubMed Open Access segment, would help address data and network sparsity issues. This would also allow the model(s) to be assessed against more commonly used tasks, such as a three-class diabetes paper classification task, and paper similarity scores derived from MESH topic labels.

BIBLIOGRAPHY

Amjad Abu-Jbara, Jefferson Ezra, and Dragomir Radev. Purpose and Polarity of Citation: Towards NLP-based Bibliometrics. Technical Report June, 2013.

Shashank Agarwal, Lisha Choubey, and Hong Yu. Automatically classifying the role of citations in biomedical articles. *AMIA Annual Symposium*, (1977):11–15, 2010. ISSN 1942-597X.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the Literature Graph in Semantic Scholar. In *NAACL-HLT*, 2018. doi: 10.1002/ardp.19723050911.

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4), 2016. ISSN 14321300. doi: 10.1007/s00799-015-0156-0.

Iz Beltagy, Arman Cohan, and Kyle Lo. SciBERT: Pretrained Contextualized Embeddings for Scientific Text. *ArXiV*, abs/1903.1, 2019.

Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

Steven Bethard and Dan Jurafsky. Who Should I Cite? Learning Literature Search Models from Citation Behavior. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 2010.

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. Content-Based Citation Recommendation. In *The Adaptive Web*. 2018. ISBN 970-676-034-2. doi: 10.18653/v1/N18-1022.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *Proc. of Language Resources and Evaluation Conference (LREC 08)*, Marrakesh, Morocco, 2008.

David M Blei, Andrew Y Ng, Michael Jordan, and John Lafferty. Latent Dirichlet Allocation. Technical report, 2003.

Kurt Bollacker, Steve Lawrence, and C Lee Giles. CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 116–123. ACM, 1998.

Lutz Bornmann and Rüdiger Mutz. Growth rates of modern science : A bibliometric analysis based on the number of publications and cited references. *ArXiv*, abs/1402.4:1–28, 2014.

Rich Caruana. Multitask Learning. *Machine Learning*, (28):41–75, 1997.

Jie Chen, Tengfei Ma, and Cao Xiao. FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. In *ICLR*, 2018. ISBN 1469-493X. doi: 10.1002/14651858.CD010013.pub2.

Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. Structural Scaffolds for Citation Intent Classification in Scientific Publications. In *NAACL*, 2019.

Isaac G Councill, C Lee Giles, and Min-yen Kan. ParsCit : An open-source CRF reference string parsing package. In *LREC*, number 3, pages 661–667, 2008.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, 2018.

L Egghe. An improvement of the h-Index: the g-index. *ISSI Newsletter*, 2(1):8–9, 2006.

Eugene Garfield. Science Citation Index – A New Dimension in Indexing. *Science*, 144 (3619):649–54, 1964.

Eugene Garfield. Can Citation Indexing be Automated? *Statistical Assoc . Methods for Mechanized Documentation*, 269:84–90, 1965. ISSN 0300-5771. doi: 10.1093/ije/dyl190.

Eugene Garfield. *The Thomson-Reuters Impact Factor*. 1994.

Mark Garzone and Robert E. Mercer. Towards an Automated Citation Classifier. In *Advances in Artificial Intelligence*, pages 337–346. Springer, Berlin, Heidelberg, 2000.

Yves Gingras. *Bibliometrics and Research Evaluation: Uses and Abuses*. The MIT Press, Cambridge, MA, 2016.

Benoît Godin. On the Origins of Bibliometrics. *Scientometrics*, 68:109–133, 2006.

Joshua T Goodman. A Bit of Progress in Language Modeling. *Computer Speech & Language*, 15:403–434, 2001.

Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. *KDD: proceedings. International Conference on Knowledge Discovery & Data Mining 2016*, pages 855–864, 2016. doi: 10.1145/2939672.2939754.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NIPS*, 2017. doi: arXiv:1706.02216v4.

Saeed Ul Hassan, Anam Akram, and Peter Haddawy. Identifying Important Citations Using Contextual Information from Full Text. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, 2017. ISBN 9781538638613. doi: 10.1109/JCDL.2017.7991558.

Myriam Hernández-Alvarez, José M. Gomez Soriano, and Patricio Martínez-Barco. Citation function, polarity and influence classification, 2017. ISSN 14698110.

Diana Hicks and Jonathan Potter. Sociology of Scientific Knowledge: A Reflexive Citation Analysis or Science Disciplines and Disciplining Science. *Social Studies of Science*, 1991. ISSN 14603659. doi: 10.1177/030631291021003003.

J. E. Hirsch. An index to quantify an individual's scientific research output. In *Proc. Natl. Acad. Sci. U.S.A.*, volume 102, pages 16569–16572, 2005. doi: 10.1073/pnas.0507655102.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, New Jersey, 2nd edition, 2009.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the Evolution of a Scientific Field through Citation Frames. *Transactions of the Association for Computational Linguistics*, 6:391–406, 2018.

Anshul Kanakia, Darrin Eide, Zhihong Shen, and Kuansan Wang. A scalable hybrid research paper recommender system for Microsoft academic. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pages 2893–2899, 2019. doi: 10.1145/3308558.3313700.

R Kneser and Herman Ney. Improved backing-off for N-gram language modeling. In *EUROSPEECH-93*, pages 973–976, 1993.

Thomas K Landauer and Susan T Dumais. A Solution to Plato's Problem : The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104(2):211–240, 1997.

Thomas K. Landauer, Peter W. Folts, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, (25):259–284, 1998. ISSN 1554351X. doi: 10.3758/BRM.41.3.944.

Ben-ami Lipetz. Improvement of the selectivity of citation indexes to science literature through inclusion of citation relationship indicators. *American Documentation*, 16(2), 1965.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proc. of the Workshop at the 1st International Conference on Learning Representations (ICLR 2013)*, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online Learning of Social Representations. In *KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014. doi: 10.1145/2623330.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. *ArXiv*, abs/1802.0, 2018. doi: 10.18653/v1/n18-1202.

Xiaobo Shen, Shirui Pan, Weiwei Liu, Yew Soon Ong, and Quan Sen Sun. Discrete network embedding. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July: 3549–3555, 2018. ISSN 10450823. doi: 10.24963/ijcai.2018/493.

Henry G. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *JASIS*, 24:265–269, 1973.

Ina Spiegel-Rosing. Science Studies: Bibliometric and Content Analysis. *Social Studies of Science*, 7(1):97–113, 1977. ISSN 14603659. doi: 10.1177/030631277700700111.

Andreas Stolcke. SRILM an extensible language modeling toolkit. In *INTERSPEECH*, 2002.

Iman Tahamtan and Lutz Bornmann. Core elements in the process of citing publications: Conceptual overview of the literature, 2018. ISSN 18755879.

Jian Tang and Meng Qu. LINE : Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, 2015. ISBN 9781450334693.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing - EMNLP '06*, 2006. ISBN 1932432736. doi: 10.3115/1610075.1610091.

Marco Valenzuela, Vu Ha, and Oren Etzioni. Identifying Meaningful Citations. *AAAI Workshop on Scholarly Big Data*, page 6, 2015. ISSN 1424404991.

Laurens van der Maaten and Geoffrey E Hinton. Visualizing Data using t-SNE. In *Journal of Machine Learning Research*, volume 9, 2008.

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying Graph Convolutional Networks. In *ICML*, 2019a.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *ArXiV*, abs/1901.0, 2019b.

Hong Yang, Shirui Pan, Peng Zhang, Ling Chen, Defu Lian, and Chengqi Zhang. Binarized attributed network embedding. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2018-Novem(Icdm):1476–1481, 2018. ISSN 15504786. doi: 10.1109/ICDM.2018.8626170.

Fattane Zarrinkalam and Mohsen Kahani. A New Metric for Measuring Relatedness of ScientificPapers Based on Non-Textual Features. *Intelligent Information Management*, 04 (04):99–107, 2012. ISSN 2160-5912. doi: 10.4236/iim.2012.44016.

Xiaodan Zhu, Peter Turney, Daniel Lemire, and André Vellino. Measuring academic influence: Not all citations are equal. *Journal of the Association for Information Science and Technology*, 2015. doi: 10.1002/asi.23179.