Deep Learning Solutions for High Expertise Domains

Sean T Yang

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

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Bill Howe, Chair

Linda Shapiro, Chair

Jenq-Neng Hwang

Jevin D. West

Program Authorized to Offer Degree:
Department of Electrical and Computer Engineering

University of Washington

Abstract

Deep Learning Solutions for High Expertise Domains

Sean T Yang

Co-Chairs of the Supervisory Committee:
Professor Bill Howe
Information School

Professor Linda Shapiro
Department of Electrical and Computer Engineering

Deep learning has had significant success in addressing big data's knowledge organization and effective communication problems. However, the technology is difficult to apply to high expertise domains due to limited accessibility to structured data. While data labeling in most deep learning problems only needs common sense, data curation in high expertise domains requires extensive knowledge and experience in these specialized domains. Thus, acquiring large-scale labeled data for high expertise domains is expensive and sometimes difficult. The scientific community is one example of a high expertise application where it is more difficult to apply deep learning due to lack of structured data. We offer solutions to communication challenges caused by an overwhelming number of publications in the scientific community. We demonstrate that scientific figures are a significant channel of communication and they can serve as a tracker of popularity and propagation of the ideas and methods. We next propose networks that automatically identify Central Figures, which are selected from the existing publications and summarize the main contributions of research papers. Central figures can be deployed on online search engines to facilitate a literature review process. We also provide evidence supporting the idea that citation behaviors in individual research documents predicts acceptance decisions, even more so than existing natural language processing models. This bibliography analysis provide additional submission reviewing strategies for publishers or conference coordinators.

We extend our studies to broader high-expertise domains based o observations from the exploration of the scientific community. First, we find that application-agnostic ontologies are often invested in these high-expertise domains. These ontologies can be utilized in Hierarchical Multi-label Classification for knowledge organization. We propose a novel framework to address multi-label classification problem and we demonstrate that the proposed model outperforms existing methods by a significant margin. We introduce Global Hierarchical Violation to measure whether the predictions follow the hierarchy constraints. We show that the current benchmarks in hierarchical multi-label classification do not properly represent the problem space and we further introduce a declarative query system to produce customizable datasets along with four benchmarks which better describe the problem.

Second, we discover that images in high-expertise domains are often equipped with short text descriptions. We present JECL, which leverages this noisy text description as a source of weak supervision. It simultaneously learns to cluster and joint representations for image-text pairs. We show that JECL outperforms existing multi-view methods on four benchmarks. The learned representations from JECL can be deployed on GraviTIE, an interactive data visualization platform that affords scalability, query, and reproducibility. It allows users to explore large heterogeneous image collections efficiently.

This dissertation offers deep learning solutions to challenges arising from low accessibility to structured data in high-expertise domains. The presented analyses within the scientific community provide strategies for researchers to communicate complex ideas efficiently. The proposed methods allow experts to organize knowledge with ontologies and to explore large-scale heterogeneous image collections with more feasibility.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

The journey toward this dissertation has been bumpy especially during a pandemic. I would have not been able to achieve this without the enormous support from the people around me.

Words can not describe my gratitude toward my advisor Prof. Bill Howe. His unparalleled support makes this dissertation possible. I am deeply thankful and inspired by his patience, guidance, and creativity. I am also extremely grateful to Prof. Linda Shaprio for her advice in the pursuit of my Ph.D. in the Electrical and Computer Engineering department. I would also like to extend my deepest appreciation to Prof. Jevin D West who inspires me with his enthusiasm for science. I would also like to thank my other dissertation committee members, Ming-Ting Sun, Jenq-Neng Hwang and John C. Kramlich for their feedback, insightful comments, and tough questions. Special thanks to the ECE advisor, Jennifer I Huberman, who provided critical advice as I moved through the program.

I would notably thank Poshen Lee who gave me the opportunities and opened doors in my early days of graduate school. I also had the greatest pleasure to work with Maxim Grechkin, Kuan-Hao Huang, Beth Roberts, John Raynolds, Luke Rodriguez, Junyi Meng, Tianyi Zhou, Abhishek Joshi, Bernease Herman, Lia Kazakova, and Bum Mook Oh. I am thankful to my friends and colleagues in UW GRAIL, DataLab and Howe's Lab.

I would also like to thank my closest friends Steven Petesch and Kyle Chao. Their kind words, encouragement, and yummy homemade Chinese food did wonders in the tough times.

I can not be where I am without my dearest family. The unconditional love and support that kept me going thorough all the ups and downs in my life. Lastly, I am thankful to my husband, Mike. This achievement would not be possible without you.

DEDICATION

To my family and to Mike

Chapter 1

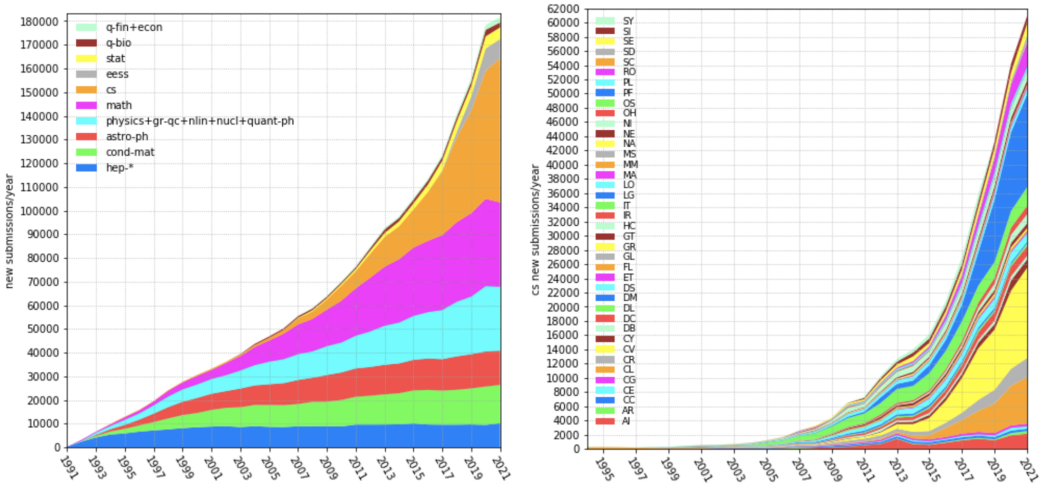
INTRODUCTION

Deep learning has had tremendous success in organizing and discovering knowledge. AlexNet [220], a first generation deep neural network, beat conventional methods on the ImageNet dataset [91] in image classification by a jaw-dropping 10.8% margin in 2012. This incident arguably kickstarted the current boom of deep learning research. High performance from deep learning models has been shown to be related to the training data size [155, 362]. Researchers can take advantage of Mechanical Turks [1] to obtain large-scale structured datasets with relatively low cost in order to address many common deep learning problems in computer vision and natural language processing. However, this is hardly achievable in high-expertise domains. Domain knowledge in such domains requires heavy investment in time and education to acquire [16]. Some examples are medical images, art history, and any scientific disciplines. Even twitter data, without knowledge in sub-culture, would seem confusing to the general public [288]. The data curation for these artifacts of knowledge production in high expertise domains, unlike labeling dogs and cats, is expensive and sometimes difficult to achieve.

Scientific documents exemplify applications that require extensive domain knowledge. The struggle from the exponential growth of the number of scientific publications is apparent in the scientific community. As shown in Figure 1.1, the submission rate per year has doubled on arXiv.org, an open-access pre-print platform. In the category of computer science, the submission rate per year has grown 80 times in the last decade and doubled in the last five years. Retrieving relevant and high quality scientific papers has become challenging for researchers [308, 120]. Moreover, the abundance of submissions also imposes significant pressure onto the publishers and reviewers. According to statistics provided by

[1] https://www.mturk.com/

Computer Science conferences [2], the number of submissions has doubled (if not tripled) in all conferences, while the reviewing process continues to be time- and resource-constrained, with average one month to two months turnaround. It has been challenging for conference chairs to manage the reviewing process and provide high quality evaluation for papers in such a short time period.



(a) (b)

Figure 1.1: The submission rate statistics on arXiv.org. The data is for 1991 through 2021 and was updated on January 3rd 2022. The figures are borrowed from arXiv.org (a) The submission rate per year has double in the last decade on Arxiv. (b) In the computer science category, the submission rate per year has grown 80 times in the last decade and doubled in the last five years.

As members of the scientific community, we want to make contributions to address the emerging issues through novel deep learning solutions. We focus on understanding how scientists communicate and make complex ideas in a research paper more accessible with figures and references. Unlike text, figures and references are not directly machine readable and are relatively under-explored as a means of facilitating scientific communication. For example,

[2] https://github.com/lixin4ever/Conference-Acceptance-Rate

to the best of our knowledge, there are 83 publications focusing on applications of text in research papers in the last five years. Only 37 publications study scientific figures and citations. We believe that figures are a better modality to communicate complex ideas, because it has been shown that humans better perceive information presented visually[389, 292]. Also, diagrammatic representations make figures extremely easy for humans to understand because of the grouped information, location indexed data, and supporting perceptual inferences [227]. These properties allow readers to avoid long searches in sequential representations (text). Citations have been studied to serve as a measure of scientific impact [33, 178] and as a marker to track scientific evolution [188, **?**]. However, few studies have focused on how referencing in individual papers communicates vital information to readers. Understanding how individual papers connect with readers through referencing can be beneficial for future researchers in adopting correct referencing behavior.

While contributions to the scientific community are significant, the generalization to other high-expertise domains is limited. To impact broader communities, we leverage our experience and observations in the scientific domain. We make two observations during our exploration of scientific communities. First, researchers tend to invest in application-agnostic ontologies to describe the relationships between phenomena in the world. These ontologies exist in a wide variety of domains on the web [355, 354], in urban settings [4, 112], finance [358, 368], oceanography [333], and art [86]. Second, images in high expertise domains tend to pair with short text descriptions providing the content or purposes of the figure, as we can see in medical imagery (paired with unstructured physician's notes), scientific figures, art imagery (paired with artist's or curator's description), or archaeological artifacts (paired with researcher's notes). These noisy descriptions, while unreliable, can be a source of (weak) supervision for learning. The solutions addressing utility of ontologies and image-caption pairs can be generalized to other high expertise domains outside of the scientific community.

This dissertation both aims to address the challenges caused by big data in high expertise domains and to facilitate efficient and effective scientific communication with deep learning methods. There are three main research objectives: (i) Investigating novel approaches to improve communication within scientific communities (ii) Leveraging application-agnostic ontologies with Hierarchical Multi-label Classification (iii) Methods for exploratory analysis

on large-scale heterogeneous image-text collections. The following describes the research objectives and outlines this dissertation.

RQ1: Investigating novel approaches to improve communication within scientific communities In the first half of this dissertation, we focus on the scientific community and use it as a case study to understand the challenges in high expertise domains. With exponential scientific publication growth, communicating complex ideas and recognizing relevant scientific documents efficiently becomes critical. While textual content has been the main means of communicating within the scientific community, figures and citations are also of significant value for scientific communication. However, figures and citations in the context of communication are rarely explored. We aim to answer the following questions in this dissertation: (i) Do different disciplines communicate differently with figures? (ii) How are the visual literature compared to communication from text and citation? (iii) How can we use existing figures from a scientific publication to better represent the main contributions of the publication? (iv) While citations have been shown to serve as a marker of scientific impact [33] , how do citations in individual research papers communicate with the readers?

In Chapter 2, we show that there are distinctive patterns of figures across sub-disciplines and that these visual patterns expose new modalities of communication that are not identifiable by either text or citation graphs. In Chapter 3, we introduce the concept of Central Figures which summarize the main ideas of the existing studies, just like graphical abstracts. These figures convey the key findings of the research and are used to summarize important results, explain the key methods, or provide additional discussion. We also train a deep learning model to automatize recognizing the central figure of each publication. In addition, we present bibliographic analysis in the context of conference acceptance in Chapter 4. We demonstrate that bibliographic references not only can improve prediction performance, but references alone can achieve higher accuracy than the state-of-the-art NLP models. This finding indicates that citations in individual papers may indirectly communicate with the readers.

RO2: Leveraging application-agnostic ontologies with Hierarchical Multi-label Classification In high-expertise domains, human attention tends to be invested in designing and curating application-agnostic ontologies rather than on hand-labeling application-specific datasets. Scientists use these graph-structured ontologies to convey their understandings of the world. With the difficulty to acquire labeled data in high expertise domains, these well curated ontologies can serve as an additional supervision for learning tasks. However, despite the opportunity for supervision, relationships among labels are often ignored. For example, the ImageNet dataset led to AlexNet [220], ResNet [151], and VGG [347], but all three models ignore the hierarchical structure of the labels. In response, we explore Hierarchical multi-label classification (HMC) problem with the objective to expand the utility for ontologies. Hierarchical Multi-label classification selects multiple labels from a label hierarchy for an input data record. Chapter 5 includes comprehensive contributions in HMC problems. We propose a novel framework, Surj, which learns offline representations of the ontology using a graph autoencoder and separately learn to classify input records, reducing dependence on training data. Our literature review reveals that no existing HMC algorithm leverages graph neural networks. We introduce a novel metric to measure hierarchy violations. Several existing HMC algorithms [390, 131] design their models to prevent hierarchy violations, but none of them evaluate their effectiveness. To address this issue, we introduce Global Hierarchy Violation to comprehensively measure whether predicted results from a HMC model violate hierarchical constraints. In addition, we find that the existing HMC benchmarks do not properly represent the problem space from distributional and structural analysis. We present a declarative query system called Onotlogue for generating custom benchmarks with specific properties, then use this system to design four new benchmarks extracted from DBPedia that better represent the problem space.

RO3: Methods for exploratory analysis for large-scale heterogeneous image-text collections. While training size is shown to link to the performance of deep learning networks [155, 362], it is expensive to curate images especially in high expertise domains. Without structured labeled data, it is challenging to analyze large-scale image collections. However, image-caption pairs arise frequently in high-value applications, such as scientific

figures and art imagery. In Chapter 6, we propose a framework that leverages these image-caption pairs and learns both representations and cluster centroids for the input datasets. The learned representations can be deployed on an online interactive visualization system, GraviTIE, which affords scalability, query, and reproducibility.

Chapter 2

DATA MINING IN VISUAL LITERATURE

2.1 Introduction

Increased access to publication data has contributed to the emergence of the Science of Science (SciSci) as a field of study. SciSci studies metrics of knowledge production and the factors contributing to this production [120]. Citations and text are the primary data types for measuring influence and tracking the evolution of scientific disciplines in this field. Dong et al. [99] use citations to study the growth of science and observe the globalization of scientific development within the past century. Vilhena et al. [380] characterize culture holes of scientific communication embedded in citation networks. However, among the studies in SciSci, the use of visualizationhas received little attention, despite being widely recognized as a significant communication channel within disciplines, across disciplines, and with the general public [238].

Humans perceive information presented visually better than textually [292] due to the highly developed visual cortex [389]. As a result, figures play a significant role in academic communication. The information density of a visualization or diagram can represent complex ideas in a compact form. For example, a neural network architecture diagram conveys an overview of the method used in a paper without requiring code listings or significant text. Moreover, the presence of a neural network diagram can be a better indicator that the paper involves the use of a neural network than any simple text features such as the presence of the phrase "neural network."

Despite the importance of the figures in the scientific literature, they have received relatively little attention in the SciSci community. Viziometrics [238] is the analysis of visual information in the scientific literature. The term was adopted to distinguish this analysis from bibilometrics and scientometrics, while still conveying the common objectives of understanding and optimizing patterns of scientific influence and communication. Lee et al. [238]

has shown the relationship between visual information and the scientific impact of a paper. In this study, we demonstrate that visual information can serve as an effective measure of similarity that can demarcate areas of knowledge in the scientific literature.

Different scientific communities use visual information differently, and one can use these differences to understand communities of practice across traditional disciplines and show how ideas flow between these communities. We consider three hypotheses: H1) Sub-disciplines use distinguishable patterns of visual communication just as they use distinguishable jargon, H2) These patterns expose new modalities of communication that are not identifiable by either text or the structure of the citation graph, and H3) By classifying and analyzing use of specific types of figures, we can track the propagation and popularity of certain ideas and methods that are difficult to discern using text or citations alone (e.g., inclusion of neural network diagrams suggest contributions of new neural network architectures).

To test these hypotheses, we extract over 5 million scientific figures from papers on arXiv.org, process the images into low-dimensional vectors, then build a *visual signature* for each field by clustering the vectors and computing the frequency distribution across clusters for each discipline. We use these signatures to reason about the similarity between fields, and compare these measures to prior work in understanding scientific community structure using text [380] and the citation graph [103, 380]. Citations and text have been used to circumscribe knowledge domains, but this is the first study that shows that figures can also delineate fields.

We compare the pairwise distances between these three matrices using the Mantel test [268], a common statistical test of the correlation between two distance matrices. We find that the visual distance is moderately correlated to citation-based metrics ($r = 0.706$, $p = 0.0001$, z score $= 5.103$) and text-based metrics ($r = 0.531$, $p=0.0002$, z score $= 5.019$). We also perform hierarchical clustering on all distance matrices to provide a qualitative comparison of the results, finding that the hierarchical structure of the fields largely agrees, but with some significant exceptions. We then consider pairs of fields that are visually distinct but similar in either text distance or citation distance, suggesting differences in the visual style of how ideas are presented. For example, we find that *Computation and Language* is visually distinct from other *Computer Science* disciplines despite being quite

similar in citation distance, because the former includes far more tables of data.

Finally, we consider specific cases of the use of particular types of figures can indicates a common method or idea in a way that text and citation similarity do not. We conduct a case study on two popular types of visualizations, neural network diagrams and embedding visualizations used to show clusters. The analysis indicates that visualizations can be used to make inferences about concept adoption within scientific communities. We also observe that the figures reveal the uptake of neural networks earlier than citation analysis, since citation counts take years to accrue. With this case study, we show the significance of visualizations in scientific literature, suggesting that the integration of figures into systems for bibilometric analysis, document summarization, information retrieval, and recommendation can improve performance and afford new applications. Our focus is in the scientific literature, but our methods are directly applicable to other domains, including patents, web pages [10], and news.

2.2 Related Work

Citations have been extensively studied and utilized as a measure of similarity among scientific publications. Marshakova proposed co-citation analysis [270] which uses the frequency that papers are cited together as a measure of similarity. Citations are also utilized to delineate the emerging nanoscience fields in [430, 244] and are applied to design recommendation systems [178]. However, citations only reveal the structural information in the scholarly literature and ignore the rich content in the articles.

Text has also received significant attention on analyzing the connection within scientific disciplines and documents, especially in citation recommendations [173, 360]. Vilhena et al. [380] proposed a text-based metric to characterize the jargon distance between disciplines. However, ambiguity and synonymity of text makes text-based model less ideal [221].

A number of studies have focused on mining the scientific figures. Chart classification was well-studied by Futrelle et al. [125], Shao et al. [344], and Lee et al. [240]. Recent studies have been focusing on the extraction of quantitative data from scientific visualizations, including line charts [259, 345], bar charts [13], and tables [113]. Researchers have also investigated the techniques to understand the semantic messages of the scientific figures.

Kembhavi et al. [195] utilized a convolution neural network (CNN) to study the problem of diagram interpretation and reasoning. Elzer et al. [109] studied the intended messages in bar charts. Several visualization-based search engines have also been presented. DiagramFlyer [68], introduced by Chen et al., is a search engine for data-driven diagrams. VizioMetrix [237] and NOA [63] are both scientific figures search engines with big scholar data, while they both work by examining the captions around the figures. We see visual-based models for demarcating knowledge domains as a next step in this area of research.

Researchers have explored other aspects of a research paper for measuring the distance between disciplines. The frequency of mathematical symbols in papers are used to delineate fields by West et. al [395], but mathematical symbols are not as ubiquitous as other components. Visual communication is a significant channel for conveying scientific knowledge, but is relatively less explored.

2.3 Experimental Setup

We describe details of the data and the method used in this subsection.

Data The data for this study comes from the arXiv. The arXiv is an open access repository for pre-prints in physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering, systems science, and economics. The variety of disciplines allows consideration of information between fields, in contrast to more specialized repositories such as PubMed. There are 1,343,669 research papers which include 5,009,523 figures on arXiv through December 31st 2017.

2.3.1 Processing Pipeline

Fig. 2.1 shows the pipeline to characterize scientific disciplines using visual information. Each step will be explained in the corresponding numbered paragraph.

Convert Figures Into Feature Vectors We first embed each figure into a 2048-d feature vector using the pre-trained ResNet-50 [151] model. The figures are re-sized and padded with white pixels to be 224 x 224 before being embedded by pre-trained ResNet-50. ResNet-

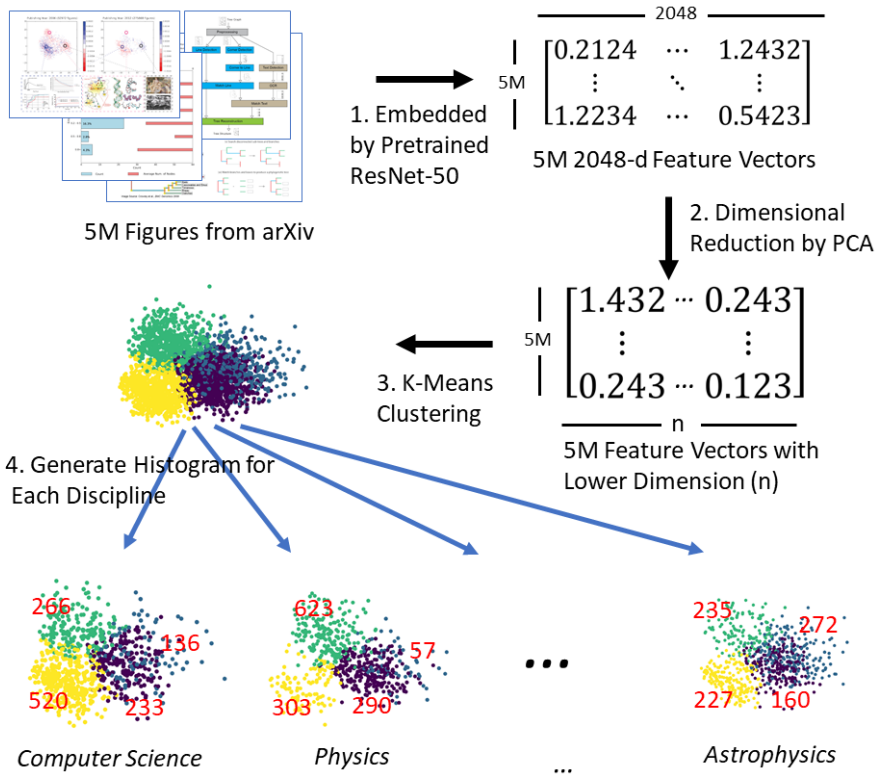


Figure 2.1: Overall pipeline. Figures are mapped to vectors using ResNet-50, dimension-reduced, then organized into a histogram for each field. The distances between these histograms are used to infer relationships and information flow.

50 was trained on the ImageNet [91] corpus of 1.2M natural images. Even though the model was trained on natural images, we find that the early layers of the network identify simple patterns (lines, edges, corners, curves) that are sufficiently general for the overall network to represent the combinations of edges and shapes that comprise artificial images as well. Although we posit that a custom neural network architecture could be designed to incrementally improve performance on artificial images, we do not further consider that direction in this study.

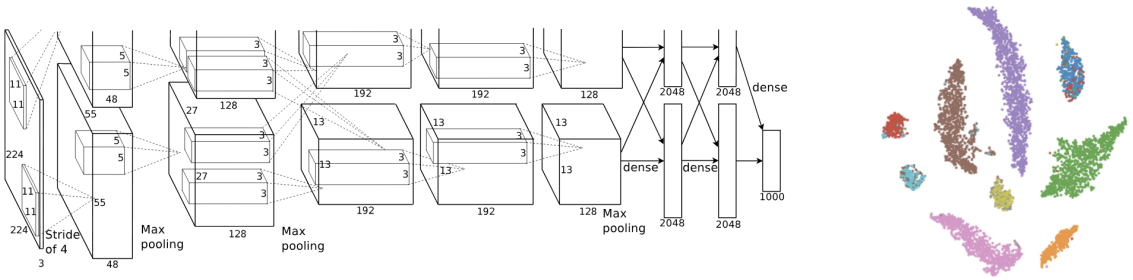
Dimension Reduction We reduce the dimension of each figure vector using Principal Component Analysis (PCA). The high-dimensional vectors produced by ResNet-50 contain more information than is necessary for our application of computing the visual similarity between fields, and we seek to make the pipeline as efficient as possible. Plus, the ResNet model is pre-trained by natural images, while scientific figures have a lot more white areas, which make the embedding vectors more sparse, than natural images. Distances tend to be inflated in high dimensional space, reducing clustering performance [32]. We follow the typical practice of applying dimension reduction prior to clustering. Our original hypothesis was that a very low number of dimensions (10) would be sufficient to capture the differences between fields, but in our evaluation the higher values (200+) produced stronger correlations with other methods of delineating fields. We considered different values of this parameter using a sample of 1.5M figures from the 5M figure corpus. The results of the experiment are presented in Section 2.4.

Cluster the Figure Corpus The distribution of different types of figures carries significant information about how the visual communication is different in each discipline and could further represent each category. We cluster our figure corpus with K-Means clustering to aggregate similar figures. Although more advanced methods of clustering could provide better results, we aim to demonstrate that the approach can work even with very simple methods. The objective of this study is to show the utility of the figures for potential applications, rather than to propose a specialized framework for specific task. The experimental results are shown in Section 2.4.

Visual Signatures for Each Discipline We cluster the figures with number of centroids $k = 4$ and generate the normalized histogram for each discipline to acquire visual signature of each discipline.

After the visual signature of each discipline is generated, we calculate the Euclidean distance between each pair of disciplines. We evaluate the computed visual similarity between disciplines by comparing to citation-based and text-based metrics described in previous work, which are explained in Section 2.3.2.

Classifying Figure Types In this section, we describe the process of training the classifier to identify specific figure types, which we will use to understand how the use of particular styles of visualization and diagrams propagate through the literature. We consider two specific examples: neural network diagrams (associated with the rapid increase of neural network methods in the literature) and clustering plots (associated with the use of unsupervised learning). Examples of these visualizations are shown in Figure 2.2.



(a) (b)

Figure 2.2: Examples of neural network diagram and embedding visualization. (a) An example of a neural network diagram. The diagram is borrowed from the AlexNet paper [220]. (b) An example of an embedding visualization. The plot is borrowed from MultiDEC paper [414].

Sethi et al. [341] characterize six different figure types to demonstrate neural network architecture. We label 10,651 figures from arXiv, which includes 1,503 neural network diagrams, 1,057 embedding visualizations, 8,091 negative examples. For neural network diagrams, we label them according to the taxonomy suggested by Sethi et al. [341], but we exclude figures in table format. We consider a figure as an embedding visualization if the figure is used to visualize the representation distribution of the data. The annotators make use of images and captions to label the images. We extract visual features from the fully connected layer of a ResNet-50 [151] model, which is pre-trained by 1M ImageNet dataset [91]. The figures are resized to 224x224 and a 2048-d numeric vector is acquired for each figure. The labeled image set is then split into training, validation, and test set with 8:1:1 ratio to

train a deep neural network (DNN) classifier. We tune the depth of the model, dimension of the layers, dropout rate, learning rate, decay ratio, and training epochs.

2.3.2 Comparison with Citation- and Text-based Methods

We use the Mantel test [268], a standard statistical test of the correlation between two matrices, to compare visual distance with the distance matrices created by (1) Average shortest citation distance [103, 380] and (2) Natural language jargon distance [380]. Citations and text have been extensively analyzed and employed to measure the similarity among research articles, and both of the measures have had success on information retrieval and recommendation systems among scholarly documents. Therefore, we consider citation distance as our benchmark of the task and text distance as an alternative comparison.

Average Shortest Citation Path We compute the average shortest path between each pair of fields as a measure of similarity. Average shortest path [103] is one of the three most robust measures [39] of network topology, in addition to its clustering coefficient and its degree distribution. Vilhena et al [380] used this method to measure distance in the citation network to compare with their text-based metric.

Average shortest path is computed as follows:

$$D_{ij} = \frac{1}{n_i n_j} \sum_{n_i} \sum_{n_j} d(v_i, v_j)$$

where n_i is the number of vertices in field i and n_j is the number of vertices in field j. The average shortest path between field i and field j, D_{ij}, is the average of all paths between all vertex pairs, v_i and v_j.

Our citation graph is obtained from the SAO/NASA Astrophysics Data System (ADS)[108], a digital library portal maintaining three bibliographic databases containing more than 13.6 million records covering publications in Astronomy and Astrophysics, Physics, and the arXiv e-prints. The creation of the citations in ADS [7] is started by scanning the full-text of the paper to retrieve bibcode for each reference string in the article, followed by computing the similarity score between the ADS record and the bibcode. The citation pairs are generated if the similarity is higher than the threshold. This data has been extensively used in several

bibliographic studies [126, 224]. There are 14,555,820 citation edges within our arXiv data corpus.

Jargon Distance We also compare our results to text metrics based on cultural information as represented by patterns of discipline-specific jargon. Jargon distance was first proposed by Vilhena et al. [380], where the authors quantitatively measure the communication barrier between fields using n-grams from full text. The jargon distance (E_{ij}) between field i and field j is defined as the ratio of (1) the entropy H of a random variable X_i with a probability distribution of the jargon or mathematical symbols within field i and (2) the cross entropy Q between the probability distributions in field i and field j:

$$E_{ij} = \frac{H(X_i)}{Q(p_i||p_j)} = \frac{-\sum_{x \in X} p_i(x) \log_2 p_j(x)}{-\sum_{x \in X} p_i(x) \log_2 p_j(x)}$$

Imagine a writer from field i trying to communicate with a reader from field j. The writer has a codebook P_i that maps the natural language or mathematical symbols to codewords that the reader has to decode using the codebook P_j from field j. A small jargon distance means high communication efficiency between two fields and are closely related. This metric could be easily applied to natural language jargon to explore how the communication varies through these two channels across disciplines. We compute the jargon distance between two different disciplines by applying the metrics on unigram from abstracts.

2.4 Experimental Results

We show that the distance between visual signatures can be used to determine the overall relationships between fields in a manner similar to prior methods, but that this approach also exposes information that prior methods cannot. In this section, we present the experimental results on picking the number of dimensions and clusters. Next, we show the capacity of visual distance to reveal the relationships across scientific disciplines by showing global agreement between visual distance and citation distance (H1). Also, we examine each cluster to understand the visual composition and find that each cluster is dominated by a certain type of visualization, extending prior work in the life sciences that used coarse-grained labeling of figure types [238]. Furthermore, we show that citation distance and visual distance

disagree in certain cases, and consider one case in particular (H2). Finally, we consider cases where the presence of a particular type of figure can indicate the use of a method or concept in a way that text and citation similarity do not in Section 2.5 (H3). We demonstrate that the figures in the scientific literature can serve as an indicator of concept adoption that travels faster than citation count.

Table 2.1: Choosing the number of clusters (k).

Dimension	Explained Variance Ratio	Average of Correlations to Citation Distance	Maximum Correlation to Citation Distance	Maximum at k=?
16	52.0%	0.661	0.737	15
32	63.7%	0.631	0.768	3
64	73.9%	0.660	0.769	4
128	82.3%	0.662	0.770	4
256	88.9%	0.672	0.793	4
320	90.7%	0.674	0.793	4

Choosing the number of dimensions and clusters Our pipeline involves two hyperparameters: the number of dimensions to retain via PCA and the number of clusters to assume when constructing visual signatures. We determine these parameters experimentally. The results of our analysis of PCA dimensions appear in Table 2.1. The explained variance ratio shows the percentage of variance explained by the selected components. The variance explained grows insignificantly after 256 components. The average correlation with citation distance shows the average of the correlations between visual distance and citation distance across all the numbers of centroid k (from 2 to 30). We evaluate our method by conducting the Mantel test [268] to compare the correlation between visual distance and citation distance. It confirms our hypothesis that the correlation increases when more components are used, but it converges after sufficient information is preserved. Maximum correlation to citation distance shows the maximum correlation of the specified dimension among different options of number of centroid k, and the k contributing the maximum correlation is shown

in "Maximum at k = ?". Surprisingly, the maximum correlation happens at a larger number of centroids with the low dimension of figure vector. Our interpretation is that there is not sufficient information preserved by low dimensional space.

We ran a second experiment to determine the number of centroids k. Initially, we expected the correlation with other measures to be higher using larger values of k, since the diversity of figures in the literature appears vast. However, considering k = 100, 200, and 400, we found that larger values of k generate lower correlations with citation distance (correlation coefficient around 0.4), due to overfitting to rare, low-confidence clusters. Lowering k to the range of 2 to 30 performed better; these results appear in Table 2.1. The relatively low values of k suggest that there are relatively few modalities of visual communication in use across fields. The maximum correlation occurred at k = 4 in most of the experiments. We further discuss the interpretation of these results in Section 2.4.

Delineating Disciplines In this section, we demonstrate the ability of visual distance to characterize the relationships between fields, quantitatively and qualitatively. Quantitatively, we conduct the Mantel test [268] with Spearman rank correlation method to compare two different distance matrices to reveal the similarity between two structures. We also perform hierarchical clustering using the UPGMA algorithm [326] to visualize the hierarchical relationships across disciplines, qualitatively. Vilhena et al. [380] used similar technique to qualitatively visualize how disciplines are delineated, but the data they used was from JSTOR, which focuses on biological science and social science so that it is not comparable with our task.

Table 2.2 shows the correlation results between different distances. The first two columns indicate the methods being compared and the Results column shows the correlations. The correlation between visual distance and citation distance ($r = 0.706$, p value = 0.0001, z score = 5.103) is higher than the correlation between jargon distance and citation distance ($r = 0.697$, p value = 0.0001, z score = 5.989), providing evidence for our hypothesis that styles of visual communication are a stronger indicator of communication and influence than the terminology used by a field. Visual distance is also moderately correlated to jargon distance with $r = 0.531$, p value = 0.0002, and z score = 5.019. This result is

expected. It verifies our first hypothesis: sub-disciplines use distinguishable patterns of visual communication. Correlation between visual distance and citation distance is sufficient enough to show that visual distance is capable of characterizing general relationships between disciplines, but it also reveals that there are still differences between citation distances and visual distance. We will elaborate the different connections visual distance expose in Section 2.4.

We then perform hierarchical clustering, using the UPGMA algorithm [326], to qualitatively visualize how different methods group similar disciplines together and separate dissimilar disciplines. The hierarchical clustering results for visual distance, citation distance, and jargon distance are shown in Fig.2.3. We observe similar patterns between visual distance and citation distance where *Computer Science*, *Statistics*, *Math*, and *Mathematical Physics* are isolated from other physics-related fields of study. There is inconsistency between visual distance and citation distance in the field of *Quantitative Biology*, which is the outlier in citation distance, but is assigned to the physics-related cluster in visual distance.

Table 2.2: The correlation results between distance matrices.

		Results
Visual Distance	Citation Distance	r = 0.706 p = 0.0001 z = 5.103
Visual Distance	Jargon Distance	r = 0.531 p = 0.0002 z = 5.019
Jargon Distance	Citation Distance	r = 0.697 p = 0.0001 z = 5.989

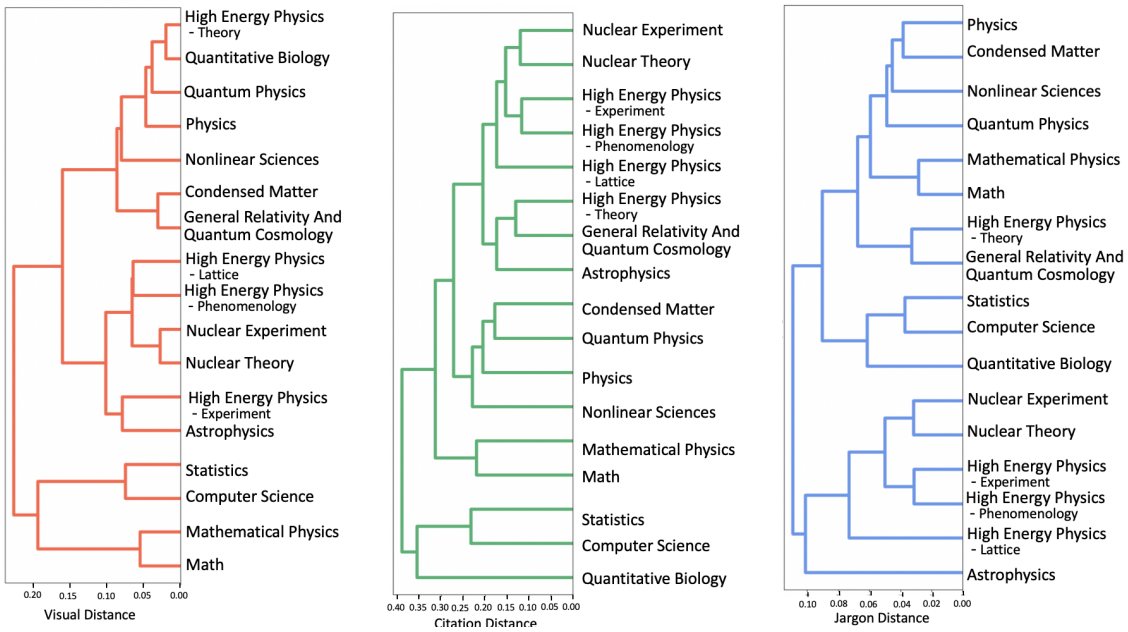


Figure 2.3: The hierarchical clustering dendrogram of visual distance (left), citation distance (middle), and jargon distance (right). Citation distance is a benchmark in our task. It shows a similar pattern to visual distance where *Computer Science*, *Statistics*, *Math*, and *Mathematical Physics* are separated from the rest of the disciplines. The inconsistency between citation distance and visual distance is *Quantitative Biology*, which is clustered with physics-related disciplines in visual distance while it is isolated in citation distance. On the other hand, Jargon distance segregates disciplines differently from visual distance and citation distance in the high level. High Energy Physics and Nuclear are separated from the rest where Quantitative Biology, Computer Science and Statistics are isolated in the sub-cluster.

Analyzing Clusters We classify the figures in each cluster to understand the visual composition of each cluster. We use the convolutional neural network classifier in [240] to categorize figures into five categories: (1) Diagrams (2) Plots (3) Table (4) Photo and (5) Equation. The classification results are shown in Fig. 2.4. Surprisingly, each cluster is prominently associated with a certain type of visualization: Cluster#0 is primarily com-

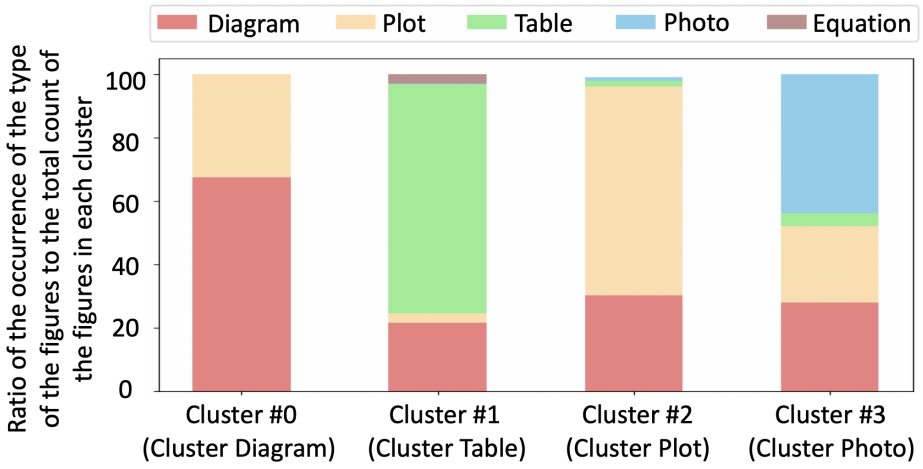


Figure 2.4: The visual composition of each cluster. It appears that each cluster has one dominant visualization.

posed of diagrams (Diagram), Cluster#1 is primarily composed of tables Table, Cluster#2 is primarily composed of plots of quantitative information (Plot, and Cluster#3 is primarily composed of photos Photo. These results corroborate previous work that used supervised methods and manual labeling to categorize figures into five classes (Diagram, Plot, Table, Photo, and Equation) [238]. The distribution of figures helps to reveal the properties of each discipline. For instance, Cluster Plot is dominant in *Quantitative Biology* (48%) and *Nuclear Experiment* (60%), which may indicate the degree to which these fields can be considered experimental and data-driven. The distribution could further be used to group similar disciplines and separate the dissimilar fields as we show in the previous section.

Visuals delineate differently than citations In this section, we focus on the cases in computer science where visual distance and citation distance disagree and we validate our second hypothesis: visual patterns expose new modalities of communication that are not identifiable by either text or the structure of the citation graph. The analysis aims to answer the following questions: (1) Where are there visual differences in the disciplinary landscape when compared to citation differences? (2) What is revealed about the fields where visual differences occur?

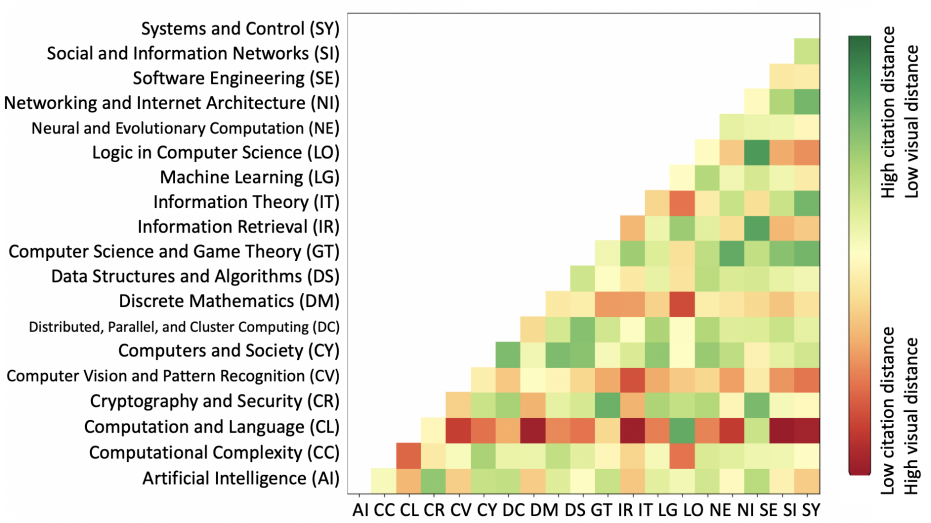


Figure 2.5: Heat map of differences between visual and citation distance. We normalize visual distance and citation distance and subtract visual distance from citation distance to expose the discrepancies. Red indicates that two subfields are visually distant but near in citation distance. Green indicates that two subfields are distinct in citation distance but visually similar. *Computation and Language* is visually different across the subfields in *Computer Science* but relatively close in terms of citation distance.

We normalize visual distance and citation distance, then subtract visual distance from citation distance to expose the discrepancies. Fig. 2.5 shows that there is a significant disagreement between visual distance and citation distance for the subfield *Computation and Language*. Red cells show the disagreements where fields are visually distinct but similar in citation distance. Green cells, in contrast, indicate disciplines that are visually similar, but far apart in citation distance. We observe that *Computation and Language* is generally close to all other categories in *Computer Science*, but visually distinct. We further examine the visual profile of *Computation and Language* in order to better understand the reasons for the divergence between these two distances.

Fig. 2.6 shows the distribution of the figure usage in *Computation and Language* (CL) and *Computer Science* (CS) over the past ten years. We make two observations from this

stacked bar chart: (1) Cluster Table dominates the visual communication style with over 50% in *Computation and Language* in 2017, compared to approximately 30% in *Computer Science*, and it has been growing over the past few years. (2) The researchers in *Computation and Language* use very few figures associated with Cluster Photo. We further investigate the reason that tables are largely used in *Computation and Language* by analyzing the cluster textually. We conduct topic modeling on the captions of the figures of Cluster Table using Non-negative Matrix Factorization (NMF) [233] with five topic numbers. In Table 2.3, we display the top 10 keywords of each topic along with the ratio of the count of the figures in each topic to the total count in the cluster over the past 10 years. We also look at the images in each topic to help us understand the purpose of each topic. Based on the keywords and the images, we can infer that Topic 0 mostly contains table with comparison data to other models, Topic 1 includes the examples of the language and words, Topic 2, which is similar to Topic 0, also involves comparing results between different models. Topic 3 consists of statistics about the dataset. Topic 4 is a mix of the tables and diagrams which mostly are used to illustrate the architecture of LSTM models. It appears that tables to compare the accuracy of different models have been growing significantly, from 46.4% (28.6% + 17.8%) in 2008 to 60% (47.6% + 12.4%) in 2017, suggesting that an empirical regime of research is dominant, perhaps due to improved access to advanced computational infrastructure, easy access to data and code, and the rapid growth of the field itself.

2.5 Fine-grained Figure Analysis

The classifier achieves accuracy of 0.902 on the validation set and 0.868 on the test set with precision of 0.741 and recall of 0.827 on neural network diagrams. The confusion matrix of the classifier is shown in Fig. 2.7. The classifier tends to misclassify flow charts, bar charts, and diagrams with multiple circles as neural network diagrams and the classifier is also often confused between embedding visualization and scatter plots (which are indeed quite similar). The classifier appears sufficiently effective at identifying neural network diagrams and embedding visualizations to conduct the following analysis.

We use the trained classifier to label 60k figures in computer science papers on arXiv and analyze the count of the neural network diagrams (Top line chart in Fig. 2.8) and the

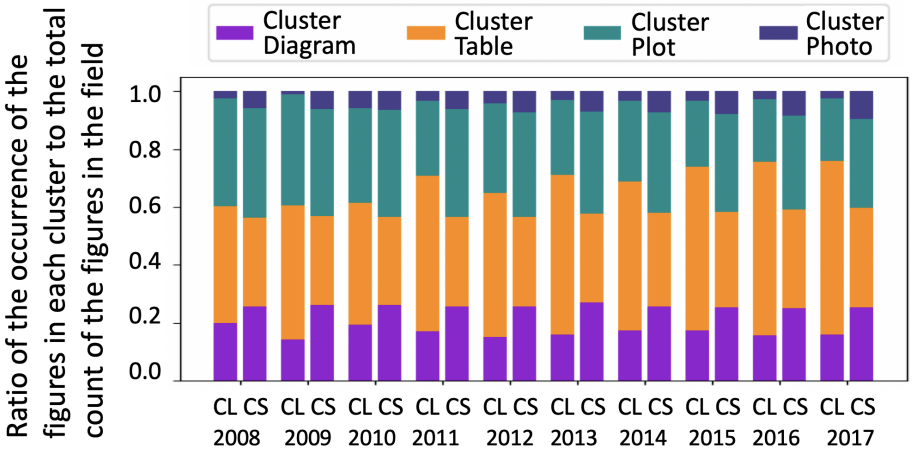


Figure 2.6: The chart shows how the distribution of the clusters evolves in *Computation and Language* and *Computer Science* over the past ten years. We could observe that Cluster Table has been growing in *Computation and Language* and researchers in *Computation and Language* use a relatively low number of figures in the Photo Cluster.

		True Label			
Predicted Label		Neural Network Diagrams	Embedding Visualizations	Negatives	Precision
	Neural Network Diagrams	43	1	14	0.741
	Embedding Visualizations	0	56	19	0.747
	Negatives	9	27	363	0.910
	Recall	0.827	0.667	0.917	Accuracy: 0.868

Figure 2.7: The confusion matrix of the figure type classifier. The classifier achieves 0.868 overall accuracy.

embedding visualizations in computer science disciplines over time. We select four categories, which are *Artificial Intelligence*, *Machine Learning*, *Computer Vision*, and *Computation Language*. These disciplines are known to be strongly involved in neural network research.

Table 2.3: Top 10 keywords for each topic in Cluster Table along with the ratio of the figure in each topic over time.

	Topic 0	**Topic 1**	**Topic 2**	**Topic 3**	**Topic 4**
	results	words	et	set	model
	table	figure	al	test	language
	models	word	2015	training	trained
	different	number	2016	data	baseline
Cluster	performance	example	2014	development	lstm
Table	best	table	2017	sets	proposed
	scores	sentence	2013	table	models
	dataset	example	results	dev	attention
	comparison	sentences	2011	used	layer
	accuracy	used	taken	statistics	performance
year	ratio	ratio	ratio	ratio	ratio
2008	28.6%	25.0%	17.8%	22.9%	5.7%
2009	31.1%	26.8%	16.1%	18.6%	7.4%
2010	31.2%	24.2%	16.9%	21.0%	6.7%
2011	34.2%	25.1%	17.2%	16.3%	7.2%
2012	39.1%	22.7%	16.0%	15.5%	6.7%
2013	37.3%	21.7%	17.3%	15.9%	7.8%
2014	39.4%	19.8%	16.0%	14.9%	9.9%
2015	43.9%	18.7%	14.2%	12.2%	11.0%
2016	45.3%	18.5%	13.4%	10.4%	12.4%
2017	47.6%	17.4%	12.4%	10.1%	12.5%

We also include *Computational Complexity*, which has less involvement in neural learning research as a control. We also compute the count of papers whose abstract include "neural network" and "deep learning" in the selected categories over time. The usage profile by field in the use of embedding visualizations is similar to that of neural network diagrams. The

trend is shown in the middle line chart in Fig. 2.8. Finally, we select six influential papers in deep learning research: AlexNet [220], GAN [132], LSTM [161], ResNet [151], RNN [329], VGG [347], and Word2Vec [278]. We calculate the received citation count of each paper for each year to show the growth of influence of these papers (Bottom line chart in Fig. 2.8). We compare these results with our visualization-based metrics to study our third hypothesis: we can use specific types of figures to track the propagation of ideas and methods in the literature.

From the three plots, we make the following observations. First, the three line charts demonstrate the same tendency: a rapid rise in recent years. It is not surprising to see this common trend; increased interest in a topic leads to both increasing citations and an increasing number of relevant diagrams across the literature.

Second, the count of papers that include "neural network" in their abstracts steadily increases from 2012 to 2014 (yellow background), as does the citation count of one particular paper, AlexNet. But there is no increase in the use of figures during this period. The cost of mentioning "neural networks" or citing a relevant paper is low, but the cost of developing a relevant figure is high. We interpret this result as evidence that the use of a figure is better correlated with the true *adoption* of a concept or method, as opposed to simply acknowledging the *relevance* of a concept or method. After a novel idea is published, the community rapidly begins to discuss the work and, potentially, cites a relevant paper. But it takes time for the community to integrate the concept into their own research. Once they have done so, the cost of developing a figure is justified, and the number of figures increases. When the concept is adopting the concept, visualizations begin to emerge in the literature.

Third, the number of neural network diagrams increases dramatically in 2015 in the four relevant disciplines, while, except for AlexNet, we do not see such rapid growth of received citation counts until 2017 (ResNet and VGG). There is a two year gap between the emergence of the use of neural network diagrams and the rise of the received citation counts. Figures, as well as text, are faster to react to the introduction of new ideas than aggregate citation counts. These results both validate the use of figures as a signal of scientific communication, but also that they expose patterns not otherwise discernible.

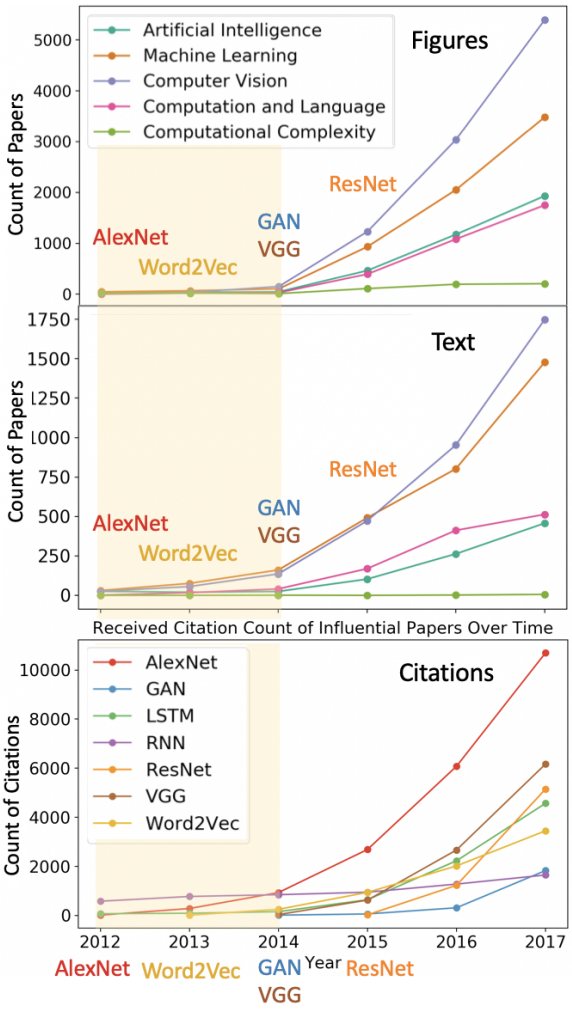


Figure 2.8: The three line charts demonstrate the trend of recent studies in deep learning using three different media: figures (top), text (middle), and citation (bottom). Top: The number of papers that include neural network diagrams over time. Middle: The count of papers that have "neural network" or "deep learning" in their abstracts over time. Bottom: The citation count of six selected influential papers in deep learning. The annotation of each influential paper indicates the publication time. Citation count of the most influential papers and use of the term "neural network" in the abstract quickly increase (yellow area), but the effect is small. The use of relevant figures increases only once authors start to truly adopt the concept in their research.

2.6 Summary

In this chapter, we lay down the foundation that scientific figures are an important channel in scientific communication. We show that figures can be as effective for differentiating communities of practice as text or citation patterns. We encode sets of images into a visual signature, then use distances between these signatures to understand how patterns of visual communication compare with patterns of jargon and citation structures. We then consider where these metrics disagree to understand how different disciplines use visualization to express ideas. Finally, we further consider how specific figure types propagate through the literature, suggesting a new mechanism for understanding the flow of ideas apart from conventional channels of text and citations.

Chapter 3

CENTRAL FIGURES IN SCIENTIFIC PUBLICATIONS

3.1 Introduction

The graphical abstract (GA), a visual summary of a scholarly article's main findings, is an emerging concept in scientific publishing. Elsevier, the largest publisher[1] of scholarly articles, requests that authors provide GAs and use them for online search results in facilitating the discovery process. With no specific guidance or requirements provided to authors, 68% and 65% of papers accepted in two of the top computer vision conferences (International Conference on Computer Vision (ICCV) and IEEE Conference on Computer Vision and Pattern Recognition (CVPR)) include "teaser figures," a form of GA. a 350% increase of graphical abstracts use in social sciences from 2011 to 2015 is demonstrated by Yoon et al. [418]. The significant increase of the use of GA can be related to the human's superior ability of perceiving visual materials. It is believed that the human's highly developed visual cortex [389] contributes to better perception of visual information than textual information [292]. As a result, visualizations play a significant role in scientific communication. With the abundance of scientific papers, GAs complement conventional text abstracts to help users quickly identify papers relating to their interests [418, 174].

Elsevier submission guidelines [2] describe a graphical abstract as a "single, concise, pictorial and visual summary of the main findings of the article" that should "allow readers to quickly gain an understanding of the main take-home message of the paper" and "encourage browsing, promote interdisciplinary scholarship, and help readers identify more quickly which papers are most relevant to their research interests" which could be a "concluding figure from the article or a figure that is specially designed for the purpose, which captures

[1] Elsevier is not the only publisher requiring GAs. Other large publishers are also requiring GAs, including Wiley-Blackwell.

[2] https://www.elsevier.com/authors/journal-authors/graphical-abstract

the content of the article for readers at a single glance.". Since not all publishing venues request a GA at the time of submission and not all authors elect to provide one, services that make use of graphical abstracts only apply to a small fraction of the scientific literature [418].

In this section, we consider the automatic selection of a "central figure" (CF) that can be used as a graphical abstract to visually summarize the paper's objectives, results, or methods, afford fast assessment of relevance, and provide a basis for new search services. This framing assumes that these CFs actually exist. To test this hypothesis, we issued 488,590 survey invitations to authors of papers on PubMed Central, asking them to identify the CF of their own publications, or indicate if no CF exists (see Figure 3.1). We also asked authors to explain the information represented in the figure to understand what role it plays. Figure 3.1 shows the survey interface. We received responses from 6,263 distinct authors across 8,353 papers. Author respondents identified a central figure for 87.6% of the papers.

Next, we use the survey responses to train a model to predict the CF in a paper. Existing GAs and teaser images are unsuitable as training data due to selection biases toward particular domains (typically visually oriented fields such as computer vision, graphics, and visualization) and because many such figures are created specifically for the purpose. We use the term central figure (CF) to distinguish from GAs. In response to publisher request, authors create GAs at the time of submission. CFs are selected from existing figures after the paper has been published. A CF may be suitable as a GA, and a GA may be identified as the central figure of a paper, but the two terms are not necessarily equivalent.

Using the results of our survey as training labels, we extract features from the figures relating to figure content, the surrounding text, and the overall paper layout. We use these features in two different models: a figure-level model that considers only one figure and its associated context at a time, and a paper-level model that considers the set of figures in a paper simultaneously. The paper-level model with features from figure content combined with the surrounding text and the overall paper layout produces the best CF identification performance. We achieve top-3 accuracy of 77.9% and exact match accuracy of 33.6% for identifying CFs with our features and model. The model outperforms heuristic baselines of selecting the first figure in the paper (25.8%), the last figure in the paper (26.9%), and

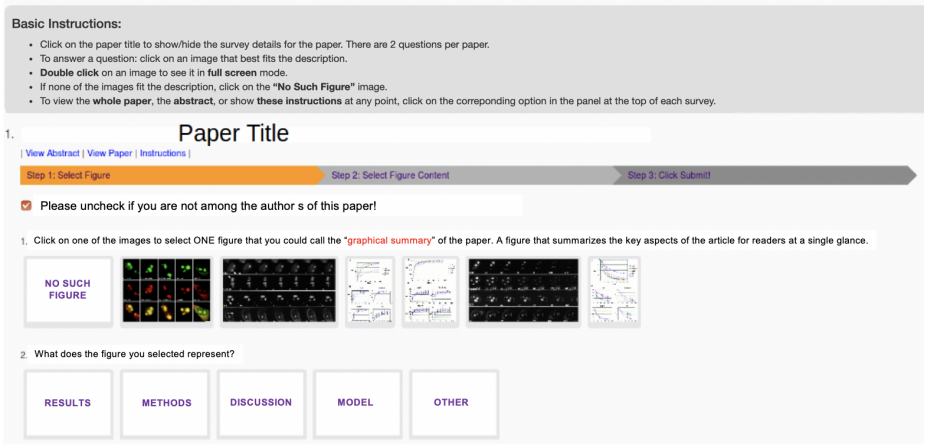


Figure 3.1: Snapshot of the survey. We asked authors of PubMed papers to identify the central figure of their own publications using this interface. Authors were asked to select a figure, if it exists, that summarizes the key aspects of the article, or choose "No such figure". We also asked authors to provide what kind of information the selected figure represents for the article from five options, which are "Results", "Discussion", "Model", "Methods", and "Other".

uniform random selection (26.4%). We find that the section title in which the figure appears and the text similarity between the abstract, the caption, and the inline reference of the figure are predictive of the CF, suggesting that authors consider these concepts in the design of their papers.

3.2 Related Work

Yoon et al. [418] investigated the frequency of graphical abstracts and the type of graphical abstracts that are adopted in social science disciplines. Hullman [174] studied the design pattern of graphical abstracts. However, only a small collection of articles were examined in both studies (772 and 54 respectively) and both studies focused on analyzing existing GAs instead of creating tools to identify GAs.

Other studies have focused on automated tools to create a representation that summarizes scientific articles have also been considered. Strobelt et al. describe DocumentCards [359], a

system to extract textual and visual content from a scientific literature and produce a high level representation. Their approach relies on simple rules to create the visual summary and can not be customized for different papers.

3.3 Data

This study was conducted using scientific papers from PubMed Central (PMC), an archive of biomedical and life science literature.

3.4 Central Figure Survey

To obtain the labeled data for CF, we launched a large-scale survey asking authors to identify CFs in their papers. We extracted email addresses from the XML files provided by PMC API and sent out 488,590 survey invitations.

Authors are asked to answer two questions for each paper:

- *Click on one of the images to select ONE figure that you could call the "graphical summary" of the paper, if one exists. A figure that summarizes the key aspects of the article for readers at a single glance.*

- *What does the figure you selected represent?*

For the first question, we used the descriptive term "graphical summary" rather than central figure to indicate our intention. Authors can select from among all the figures in the paper or select "No such figure." The latter option allows us to validate whether or not the CF is a recognizable concept in the current scientific literature. For the second question, authors may select from five options: "Results", "Discussion", "Model", "Methods", and "Other".

Survey Results As of December 1st, 2018, we had collected data on 8,353 distinct papers, from 6,263 distinct authors. Some authors provided responses for more than one of their papers, and some papers generated responses from more than one of its authors. The publishing year distribution is shown in Figure 3.2. 74.0% of evaluated papers are published after 2010. Only 12.4% (1,036) of the evaluated paper were indicated not to have a figure

that satisfies our definition of CF (890) or for which multiple authors selected different figures (146). For the remaining 87.6% of the papers, the authors identified a single CF, suggesting acceptance of the concept.

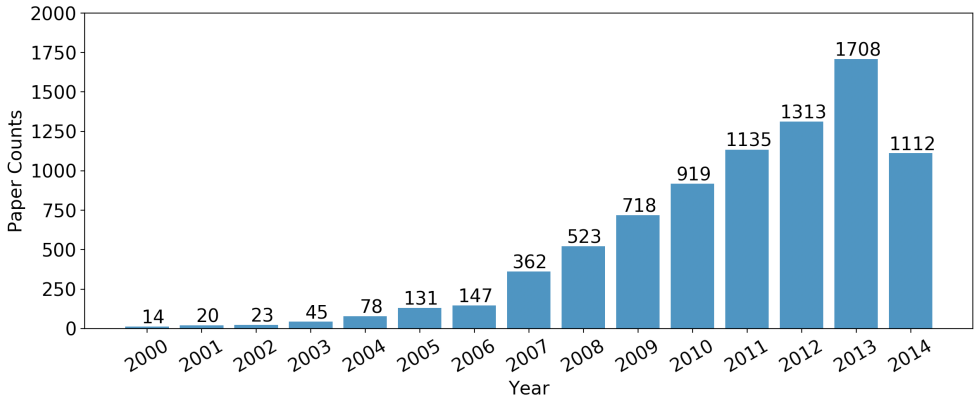
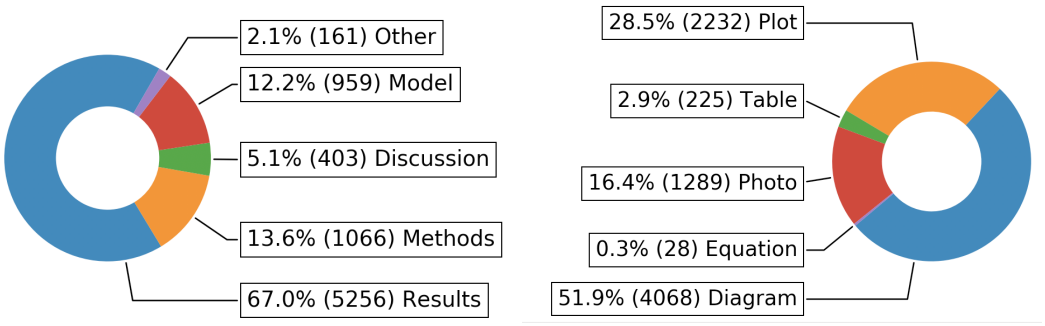


Figure 3.2: The publishing year distribution of evaluated papers. 74.0% of evaluated papers are published after 2010.

Analysis of Objectives of Central Figure Figure 3.3a illustrates the purpose of the central figure. In 67.0% of the papers, the central figure represents results, corroborating Yoon et al. who found that graphical abstracts are most frequently used to present results [418]. This use of the central figure affords an interpretation that a paper is a delivery vehicle for one main result, which supports the idea toward a results-oriented publishing model, where the unit of publishing is a scientific workflow [29, 302] or a nano-publication [281]. Methods and model were the next most popular categories at 13.6% and 12.2%. Discussion is responsible for only 5.1% of central figures. In 2.1% of the papers the authors indicated the content as Other.

Analysis of Figure Content of Central Figure After collecting the survey results, the next step is to analyze the content within the figures. Using the class assignments compiled by Lee et al. [239] and classifier approach described by Lee et al. [240], we train a classifier to identify different figure types. The training dataset, including 1871 equations,



(a) (b)

Figure 3.3: (a) Author-indicated objective of the central figures. The survey results reveal that the central figures are used to represent scientific results. (b) Pie chart of figure type distribution of central figures. 51.9% of central figures are diagrams.

3347 photos, 2849 diagrams, 2193 tables and 4680 plots, is split into training set, validation set, and test set with 8:1:1 ratio. We finetune a pre-trained ResNet-50 [151] and obtain similar model performance reported by Lee et al. [240]. We label the central figure as one of five figure types. The totals are shown in Figure 3.3b. 51.9% of the central figures in the evaluated papers are diagrams, which agrees with the findings of Yoon et al. [418] that most GAs are diagrams. We found that despite the fact that plots (graphs) and tables can both be used for presenting data, plots are much more popular when it comes to presenting key information. This result agrees with Cleveland et al. [76], who showed that fractional graph area (FGA) increases as one moves from social to mathematical and then to natural science. This finding also agrees with results from Smith et al. [349] that suggest that technical fields of science tend to use more graph-oriented figures than table-oriented. The fact that equations are rarely found among central figures is consistent with findings by Fawcett and Higginson [115].

3.5 Model Central Figures

The next task is to train a model to select the CF from all figures in a paper. We consider three sources for identifying the CF: (1) the content of the image, (2) the text describing

the image (and as it relates to the abstract), and (3) the location of the figure in the paper. We will elaborate on each source in the following subsections.

3.5.1 Image Content

The content of the image itself is not a good predictor of centrality, as we will show, since many figures in a paper look alike and our training set is limited. However, we find it useful to consider the broad type of the image as a feature. We classify each image into one of five categories, diagram, plot, table, equation, and photo, using the classifier developed by Lee et al. [239]. We label all the figures in the datasets by running the figure type classifier mentioned in Section 3.4. This categorical feature is encoded in a 5-d one hot vector to represent the visual content of the figure.

3.5.2 Text Features

Each figure is described in both a caption and in one or more inline references in the body of the paper. While both sources of text can be used as features alone, we also consider the similarity of these excerpts to the abstract as an indicator that the text serves as a summary of the overall paper. We will first explain the process of extracting surrounding text of a figure from the paper and then describe the similarity measures.

Text Extraction: We collect captions of the figures from PubMed. To extract inline references in the body of the paper, we use Science Parse [3] to parse the papers in pdf format provided by PubMed and obtain the full text in structural form. We then search the pattern that consist of words, including *Figure*, *Fig*, and *Table*, followed by a number using regular expression. We select the paragraph blocks contain the inline references in between two break line ($\backslash n$) characters. Finally, we match the index between the inline references and the captions of the figures.

Similarity Between Caption and Abstract: An abstract is a summary of the paper's results. High similarity between a figure's caption and the paper's abstract would therefore indicate that the figure plays a potential summarizing role as well. We experiment with

[3] https://github.com/allenai/science-parse

three different similarity measures: (1) TF-IDF, (2) Elmo-avg and (3) Elmo-DynaMax.

- **TF-IDF + Cosine Similarity**: We preprocess the captions, the inline references and abstracts from training set by tokenizing the documents and removing the stop words. We pick the most frequent 1,024 words to construct TF-IDF weights and the weights are acquired from the preprocessed training set. The dimensionality is picked to match with the competitive similarity measures. For each image, we apply the weights to the concatenation of the caption and the inline reference. Every abstract is also embedded in the TF-IDF vector. We finally compute the cosine similarity between the two vectors. We will refer this similarity measure as TF-IDF for simplicity.

- **Elmo-avg** [312]: Elmo is one of the state-of-the-art contextualized word embedding models. The word representations are functions of the internal states of a bidirectional language model. Elmo has been trained in large-scale scientific documents from PubMed, making Elmo a natural candidate for our task. The contextualized word representation is obtained from the top layer of the pre-trained Elmo model, and we average the word representations to acquire the representation vector for both the image descriptions and abstract. The cosine similarity is computed between the averaged word vectors of image descriptions and abstract.

- **Elmo-DynaMax** [312] [425]: Zhelezniak proposed a similarity measure, DynaMax, that dynamically extracts max-pool features based on the sentence pair. This method outperforms current baselines on several tasks [425]. The DynaMax similarity is computed between the image descriptions and the paper's abstracts from Elmo word vectors.

3.5.3 Layout

We produced two numerical features and one categorical feature from image position: (1) normalized section index, (2) image order, and (3) section heading:

- **Normalized section index**: Normalized section index is used to represent the position of the image within the layout of the paper. For example, in a paper with sections

"Introduction," "Methods," and "Results," the corresponding sequentially increasing section identifiers would be 0, 1, and 2. The normalized version of the identifiers would be its original value divided by the maximum identifier value.

- **Image Order**: The sequentially increasing numerical identifier for an image based on its order of occurrence in a paper.

- **Section Heading**: The survey shows 67% of the cases with central figures are used to represent results. To capture this feature, we constructed unigram representations of the section headings of papers in our dataset for both the entire headings and their distinct words. We then transformed the top ten frequently occurring words in the section headings unigram model in to ten unique boolean classification features, each denoting "1" for whether the corresponding word occurred in a given section heading, and "0" otherwise.

3.6 Models

In this section, we illustrate two different models to identify central figures.

3.6.1 Figure-level Model

This approach attempts to predict whether an individual figure is a central figure without considering the other figures in the paper. Let $X = \{x_i : i\}$ be the features of the images and each image corresponds to a label y_i, where $y_i \in \{-1, 1\}$. central figures are labeled as 1 and non-central figures are labeled as -1. We learn a mapping function $f : X \to Y$ using machine learning techniques, which include logistic regression, random forest, gradient boosting, support vector machine (SVM), and neural networks.

To pick the central figure from a paper $A = \{a_j : j\}$, we select the figure with highest probability predicted by each classifier f: $C_j = \arg\max_{x_i \in A_j}(P(f(x_i) = 1))$

3.6.2 Paper-level Model

This approach predicts the position of the central figure given all figures in a paper. For example, if a paper has 10 figures, we concatenate all 10 feature vectors, and then predict an integer 0..9 to indicate which figure is the central figure. Let $V = \{v_j : j\}$ represent a feature vector for each paper. v_j is a n x d vector where n is a parameter and d is the dimension of image feature. Since there are variable number of figures in different papers and basic machine learning models only take fixed dimension inputs, we introduce a hyperparameter n to serve as the threshold for the number of figures. We pad zero if the number of figure is smaller than n in a paper. For the case where the number of figure is larger than n, we select n figures whose captions are most similar to the abstract based on our TF-IDF model to fill v_j. The classifiers f will learn a mapping function $f : V \rightarrow I$, where $I \in \{0, 1, ..., n\}$ is the index of the central figure. We experiment on the ensemble and regression learning methods plus neural networks listed in previous sub section.

3.7 Experiments

In this section, we first define evaluation metrics on our task. Next, we explain the implementation details of our models. Baseline models are next introduced as comparisons. Finally, quantitative results of our image-based model and paper-based model are presented.

3.7.1 Evaluation Metrics

The image accuracy is applied to evaluate the image based model. The image accuracy is defined as:

$$imageACC = \frac{\text{True Positive + True Negative}}{\text{Total number of the images}} \tag{3.1}$$

We use two metrics, ACC and ACC@3, to evaluate the overall capability of selecting central figure from a paper.

$$ACC = \frac{N_c}{N_t} \tag{3.2}$$

where N_c is the number of the papers with correct central figure prediction and N_t is the total number of the papers.

$$ACC@3 = \frac{N_c@3}{N_t} \tag{3.3}$$

where N_c is the number of the papers with the correct central figure prediction, within the 3 figures with highest probability. N_t is the total number of papers.

3.7.2 Implementation Details

We remove the evaluated papers which do not have central figures and split the data into training, validation, and test set with 8:1:1 ratio. We run our experiments on the training and validation set. The final model is trained by the data from both training set and validation set and accuracy results reported below are conducted on test set.

Regression and ensemble models are trained using Scikit-learn and we use default values for hyperparameters. The neural network model include three fully connected layers with dimensions 100-100-n. Drop out layers with drop out rate 0.2 are inserted between the fully connected layers. All the models are trained with learning rate 0.01 and 0.01 decay for 100 epochs.

3.7.3 Baseline Models

We introduce three naive baseline models as comparisons.

- **Pick First**: The first image is selected as prediction in this model. We pick first three images in the paper as top three guesses for ACC@3 evaluation metric.

- **Pick Last**: The last image is selected as prediction in this model. We pick last three images in the paper as top three guesses for ACC@3 evaluation metric.

- **Randomly Select**: We randomly select an image as prediction. Three images are randomly selected as the top 3 guesses for the ACC@3 evaluation metric.

The performance of the baseline models is shown in Table 3.1. There are 4.68 images in a paper on average in the dataset. Accuracy of 0.264 from randomly select model makes sense.

Table 3.1: Performance of baseline models.

Pick First		Pick Last		Randomly Select	
ACC	ACC@3	ACC	ACC@3	ACC	ACC@3
0.258	0.704	0.269	0.679	0.264	0.706

3.7.4 Image-level Model

Table 3.2: Image accuracy (Equ. 3.1) of central figure classification from image-based models.

Logistic Regression	Random Forest	Gradient Boosting	SVM	Neural Networks
0.626	0.616	0.621	0.673	0.684

Table 3.2 shows the classification results from each classifier on identifying central figure. Overall, every classifier is able to achieve more than 60% accuracy on classifying between central figure and non-central figure on figure level. The accuracy of the central figure prediction given paper is shown in Table 3.3. Not surprisingly, this simple image-based model does not perform well on selecting the central figure from a list of figures. The model is not able to learn the structural relationships between figures from the same paper.

3.7.5 Paper-level Model

We run an experiment to determine hyperparamter n (the threshold for the number of figure to be accommodated for the input V). The experimental results are shown in Figure 3.4. The blue line, which corresponds to the y axis on the left, is the accuracy of the model and the red line shows the percentage of central figures that were left out because of our selection of n. The selection of n has insignificant influence to the model when n is larger than 6 and the maximum number of figure a paper has in our validation set is 12. Thus, we pick $n = 15$

Table 3.3: The results of paper-level model with different classifiers. Surprisingly, logistic regression outperforms random forest and gradient boosting. Textual content is the most useful feature on recognizing central figure, compared to visual content and the position feature.

			Logistic Regression		Random Forest		Gradient Boosting		SVM		Neural Network	
Text	Visual	Layout	ACC	ACC@3	ACC	ACC@3	ACC	ACC@3	ACC	ACC@3	ACC	ACC@3
Figure-level model												
TF-IDF	v	v	0.140	0.691	0.248	0.703	0.126	0.685	0.126	0.690	0.142	0.688
Paper-level model												
TF-IDF	-	-	0.302	0.764	0.318	0.724	0.314	0.760	0.314	0.756	0.311	0.718
-	v	-	0.289	0.730	0.278	0.693	0.286	0.731	0.319	0.748	0.282	0.724
-	-	v	0.296	0.757	0.284	0.723	0.284	0.741	0.299	0.746	0.284	0.756
TF-IDF	v	-	0.317	0.757	0.292	0.712	0.295	0.764	0.322	0.749	0.276	0.716
TF-IDF	-	v	0.323	0.782	0.277	0.690	0.335	0.765	0.273	0.708	0.280	0.750
-	v	v	0.312	0.742	0.267	0.703	0.302	0.720	0.300	0.724	0.293	0.739
Elmo-avg	v	v	0.329	0.771	0.262	0.670	0.314	0.771	0.299	0.729	0.299	0.738
Elmo-DynaMax	v	v	0.330	0.769	0.285	0.679	0.321	0.778	0.299	0.733	0.282	0.739
TF-IDF	v	v	**0.336**	**0.779**	0.267	0.701	0.314	0.760	0.302	0.727	0.306	0.745

for the rest of the experiments.

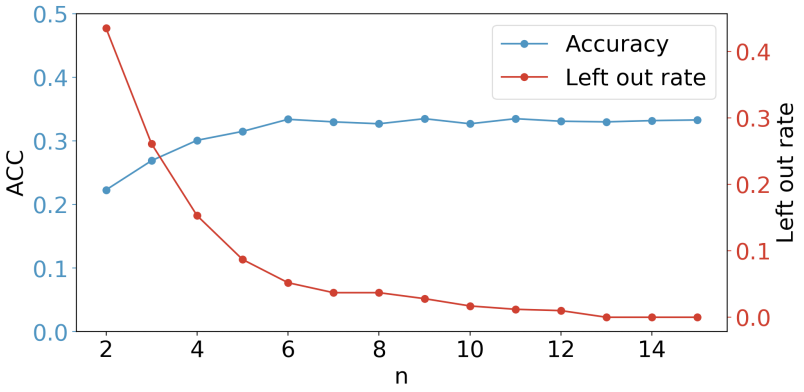


Figure 3.4: Experimental results on hyperparameter n. When n is larger than 6, selection of n does not affect the accuracy of the model.

The results for the paper-level model with different feature combinations are shown in

Table 3.3. The logistic regression classifier performs the best among all the models, including neural networks. The poor performance from neural networks is likely due to insufficient data and low-dimensional features. The text context has the most predictive power among the three sets of features, while the visual figure content has the least. Our interpretation of these results is that similarity between the figure caption and paper abstract not only provides the representation of the image but it also suggests the relationship to the paper. On the other hand, without any further information of the paper, figure type is irrelevant to determine central figure in this generation of the model. Also, surprisingly, the simple TF-IDF representation produces better performance than the Elmo word representations in more than half of the models. We speculate that the terms used in captions, and the contexts in which they are used are sufficiently specialized to allow the simpler representation to outperform pre-learned representations based on a larger corpus of text. Using max-pool followed by fuzzy Jaccard index to compute similarity between two documents has superior results over averaging the word vectors, which agrees with the findings of Zhelezniak et al. [425]. Considering the difficulty of the task and the variability of scientific figures, we see our results as a reasonable start for automatically identifying central figures.

We investigate the effectiveness of our feature selection. We replace the similarity between abstract, caption and inline reference with a text representation vector of the caption and inline reference. We also experiment with using image embedding extracted from pre-trained ResNet-50 [151] instead of categorical feature based on the figure content. Even though the model was trained on 1M natural images, we find that the embeddings that capture visual patterns and colors are sufficiently general to represent the combinations of edges and shapes that comprise artificial images as well. The results are given in Table 3.4. The experiments demonstrate that using similarity between abstract, image caption, and inline reference boost the central figure identification performance and that the categorical label of image content is more beneficial than image embedding. Our interpretation is that the high similarity between a paper's abstract and a image's surrounding text does indicate the centralization of the figure and that the original text representations and image embeddings are too sparse and noisy for the model to learn an effective function.

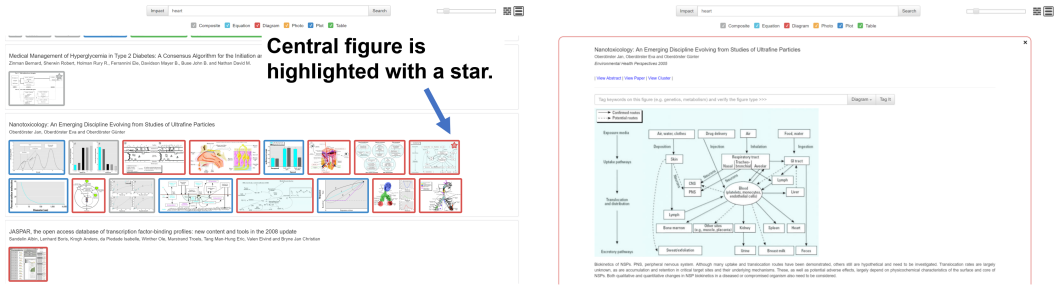
Table 3.4: Experimental results on using text representation and image embedding. *Sim()* indicates the model uses similarity between paper's abstract, image caption, and inline reference computed by the text representation in the parenthesis. *Vec()* implies the model utilizes the representation vectors derived from the model in the parenthesis. *Label* represents the model includes the categorical label described in Section 3.5.1 as image content feature. We can observe that using similarity and the categorical label of image content produces better performance than using representations.

			Logistic Regression	
Text	Visual	Layout	ACC	ACC@3
Sim(TF-IDF)	Label	v	**0.336**	**0.779**
Using Image Embedding from Pre-trained ResNet-50				
Sim(TF-IDF)	Vec(ResNet)	v	0.323	0.761
Using Text Representation Vectors				
Vec(TF-IDF)	Label	v	0.310	0.722
Vec(Elmo-avg)	Label	v	0.300	0.723
Using Both Text Representation Vectors and Image Embedding				
Vec(TF-IDF)	Vec(ResNet)	v	0.288	0.741
Vec(Elmo-avg)	Vec(ResNet)	v	0.293	0.705

3.8 Discussion

Scholarly communication is moving away from just a simple PDF. Individual insights, experiments, and conclusions can be communicated across different media and platforms. In this study, we focus on the role that visual information plays in communicating the key results, models or concepts. The idea behind a central figure is that it provides an alternative access point to the content of the paper. In some papers, it can reveal the key results and conclusions better than the title, abstract, keywords or authors. Figure 3.5 shows two proto-

types of how to introduce the central figure in an image-oriented scientific search interface, viziometrics.org. As shown in Figure 3.5(a), the central figure is highlighted with a star on the search interface. The entry page could feature the central figure along with textual abstracts as shown in 3.5(b). With these two additional features, users are able to quickly ascertain the overall concept of the article with the help of central figure at a single glance.



(a) (b)

Figure 3.5: Prototype interfaces on viziometrics.org allow individuals to search for images from scientific literature with the aid of "central figures". (a) Central figure is starred for easy recognition on searching interface. (b) Prototype of entry page for each article. The entry interface of each article could be led with the central figure along with textual abstract to help the users understand the articles quickly.

Publishing culture has changed dramatically over the last few decades due to the introduction of multiple open access platforms, such as arXiv and PubMed, as well as the significant increase of scientific publications. With more open access platforms available, the accessibility of innovative ideas pushes the advance of science and allows the community to share and communicate ideas in different formats. The presentation of new scientific ideas is no longer restricted to traditional document copies or digitized pdf formats. For example, we can easily find comprehensive ablation studies of the state-of-the-art deep learning models on GitHub. Google AI [4] hosts a blog to introduce and advertise their progress on innovative

[4] https://ai.googleblog.com/

scientific findings and technologies. Plus, the overwhelming scale of scientific publications [229, 45] that are published every year. Several recent studies [418, 174] have explored new measures for the community to quickly grasp the main messages of the scientific documents. The scientific publishing enterprise has shifted to be more open to the public and less restrictive on format, and we believe the identification and extraction of central figures can play an important role in the evolution of the scientific communication. A central figure provides a visual summary of the key results, objectives, or methods of a paper. It is adaptable to varying media and platforms, easy to share, and information-rich. We can see each central figure as a visual "nanopublication" [281] and use it to reduce the redundancy of traditional publications. Every central figure is a module of condensed ideas and can be transmitted and shared easily. Therefore, central figures can contribute greatly in the evolution of scientific communication with quick idea transferring and flexible publishing platforms.

3.9 Summary

Visualizations will play an increasingly important role in scholarly communication. The goal of this section was to focus on visual objects that convey the central findings of a research paper. We collected more than eight thousand labeled data for central figure identification from a large-scale survey. 87.6% of the evaluated papers included a central figure noted by the authors. This was evidence that central figures exist and they perform a function in scholarly communication. We extracted features from the figure content, surrounding text, and the overall paper layout as a way of training a figure-level model and a paper-level model. The results reveal that the paper-level model with all features produce the best performance overall in identifying central figures. We achieve top-3 accuracy of 77.9% and exact match accuracy of 34%. We also demonstrate that the caption, inline description, and layout shows higher importance than figure content in this task. Survey data and code are publicly available [5], and we hope the released data can attract the community to investigate this problem and further contribute to the scientific communication.

[5] https://github.com/viziometrics/centraul_figure

Chapter 4

BIBLIOGRAPHIC ANALYSIS IN RELATION TO ACCEPTANCE DECISION

4.1 Introduction

The scientific literature is doubling nearly every nine years [377]. This growth rate and its implications for scholarly communication and policy have been well-studied [229, 45, 398]. But the effect of this growth on the peer review process remains unclear. How fast are submissions growing relative to published papers? What are the consequences of this growth rate on quality control? What role, if any, can automation play in peer review as growth rates likely continue to rise? What factors influence acceptance and rejection in the context of this growth?

Over the last few years, publishers have begun to open up peer review. Computer science is leading the way. According to the statistics provided by conferences [1], the number of submissions have doubled (if not tripled) in all five conferences, while the reviewing process continues to be time- and resource-constrained, averaging a 1 month to 2 month turnaround. It has been challenging for conference chairs to manage the reviewing process and provide high quality evaluation for the papers in such a short time period.

Incorporating artificial intelligence (AI) into the review process has been proposed to address this strain on peer review [152, 51, 3], but a shift toward automation is fraught. Besides the social and policy implications, a key technical obstacle is the dearth of labeled data. However, the transparency movement in peer review is beginning to change the situation [318, 218]. OpenReview[2], for example, is a platform that several computing conferences are adopting for their peer review process. All the submitted drafts and reviews are available for a general audience to browse and download. Kang et al. [191] recently organized Open-

[1] https://github.com/lixin4ever/Conference-Acceptance-Rate

[2] https://openreview.net/

Review data into a dataset they call PeerRead, as a way of exploring automatic evaluation of scholarly documents. This included papers from ICLR and research conferences in natural language processing (ACL and CoNLL[3]). The data includes paper drafts for all submissions, reviews, and the final decisions for these conferences.

Kang et al. [191] combine metadata features, bag-of-words representations and Glove embeddings [311] of the abstract to predict the acceptance decision and achieve accuracy of 65.3% for the ICLR2017 dataset. Ghosal et al. [130] proposed DeepSentiPeer and improved the accuracy by 6% by incorporating review sentiment. DeepSentiPeer's attempt to automate the decision making process from the reviews is promising, but the problem of reviewing a significant amount of submissions still remains.

Existing work focuses on the content of the papers and the text of corresponding reviews, but we hypothesized that acceptance is strongly influenced by both the content and the references. The references of a paper indicate an author's awareness of important results, cognizance and appreciation for community norms, and connectivity between important ideas. Anecdotally, reviewers consider the references carefully in judging a paper's merit, whether by looking for specific related papers that may have been missed or as a proxy for general mastery of related topics. The relationship between the citing references in a paper and the final acceptance decision has mostly been ignored, but we contend that this structural information – how a scholarly community is connected via references – is an important feature both to improve accuracy of the models and to study the underlying processes that influence reviewing decisions.

In this study, we consider whether accepted papers and rejected paper differ in the number, recency, and venue of their references. To answer this question, we extract reference features from submitted papers and train machine learning models to predict acceptance decisions. *Surprisingly, we find that simple properties of the set of references (e.g., venue and publication date) better predict acceptance than the state-of-art NLP approaches.* We also show that combining these features with NLP methods yield a family of new state-of-the-art models that outperform models trained on either set of features alone. This result may

[3]The corresponding full names of the abbreviation of the conferences can be found in Appendix.

have significant implications for the review process: not only are references an influential feature, they may dominate the decision process! A decision process strongly influenced by a paper's references suggests that community cohesion (and potentially, insularity) may be the dominant factor in scientific communication. The findings demonstrate both the potential of references as a signal of quality but also suggest new methods to understand the strengths and weaknesses of the peer review process itself. The data and the code will be publicly available.

4.2 Related Work

We consider related work in peer review, mining the PeerRead dataset, and reference analysis.

Peer Review The efficiency, consistency, and the appropriateness of peer review has been an active discussion in the literature [317, 122, 40, 163]. The NIPS (now NeurIPS) experiment [317, 122] is one of the most well known studies in recent years. In the experiment, more than half of the papers accepted by one group of reviewers were rejected by the other group. The experiment exposed the arbitrariness of the peer review process. John Bohannon submitted an obviously flawed scientific paper[4] to 304 open access journals with supposed peer review, and 60% of the journals accepted the paper [40]. The reliability of peer review in some venues have been challenged. The response to these challenges has been increased transparency. Hojat et al. [163] revealed several biases in peer review and proposed an increase in awareness of these pitfalls. With the advance of the natural language processing, computer scientists have attempted to the automate peer review process. Ghosal et al. [130] took the first step by incorporating the sentiment of the reviews with the representation vectors of papers and reviews to predict the acceptance decision. While effective, the utility is limited because the tool requires the reviews and sufficient amount of the data.

[4]We should note the ethical issues of these hoax studies. We do not encourage this kind study as a way of studying peer review or pointing out its limitations. We find these studies dubious at best.

Mining PeerRead dataset Since Kang et al. [191] published the PeerRead dataset, several studies have demonstrated the diverse utility of this dataset. Hua et al. [170] collected peer reviews from PeerRead and other additional venues. They further annotated the sentences identifying argumentative propositions. The authors then automated the argumentative proposition detection process and found that the usage of propositions varies across venue in terms of amount, type, and topic. PeerRead was also extended to FullTextPeerRead [180] to contain the connection between the full text and cited reference and its metadata. Jeong et al. uses this extended dataset to develop a context-aware citation recommendation model. As mentioned above, Ghosal et al. [130] proposed DeepSentiPeer to predict acceptance and recommendation scores from the sentiment of the reviews and the representation vectors of the reviews and the papers. Of the work conducted so far on this data, no studies to date have focused on the references. We find the references to be highly valuable when making acceptance predictions.

Bibliography Analysis The importance of references have been studied in different contexts [272, 335, 365]. Recently, researchers have used the bibliography to predict the scientific impact of a paper [98, 429, 149, 419]. In these studies, the researchers usually use the number of references or the average citations of referred publications to represent the quality of references. He et al. [150] developed a temporal citation embedding to understand the changing roles of the publications. Boyack et al. [47] examined in-text reference to characterize the differences of citation pattern across disciplines and time. To our knowledge, there is no existing work directly investigating the influence of referencing behavior and the relationship between citation pattern and acceptance decisions.

4.3 Dataset

In this analysis, we focus on the collection of papers from the International Conference on Learning Representations (ICLR). We chose this corpus for its representation of computer science, the quality of papers, and, importantly, the availability of both accepted and rejected papers. ICLR is a globally renowned conference on artificial intelligence, machine learning, and more specifically, representation learning. As of the writing of this dissertation, ICLR

has h5-index [5] score of 150, ranked second in Artificial Intelligence on Google Scholar Metrics [6]. For comparison purpose, Neural Information Processing Systems (NeurIPS) has h-index score of 169 and the International Conference on Machine Learning (ICML) has an h-index of 135). The reviewing process of ICLR has been available online since 2013 [7]. The original papers, the peer reviews, and the rebuttals are all publicly available.

Kang et al. [191] collected paper drafts and reviews of ICLR2017 from OpenReview and created the PeerRead dataset. In total, the dataset contains 427 papers. The distribution across accepted and rejected papers is shown in Table 4.1. However, when trying to reproduce the models in PeerRead, there was high variance on the results due to limited size of the dataset. To make our analysis more generic and avoid overfitting, we crawled the papers of ICLR2018 and ICLR2019. This larger dataset is described in Table 4.1.

Venue	**#Papers**	**#Accept**	**#Reject**
ICLR 2017	427	172	255
ICLR 2018	948	336	612
ICLR 2019	1501	502	999
Total	2876	1010	1866

Table 4.1: Stats of ICLR from 2017 to 2019

Venue	**#Train**	**#Val**	**#Test**	***Total***
ICLR 2017	349	40	38	427
ICLR 2018	761	95	96	948
ICLR 2019	1203	150	151	1501
Total	2313	285	285	2876

Table 4.2: Data distribution over train, validation, and test sets.

In total, there are 2,876 papers, including 1,010 accepts and 1,866 rejects. We do not include any workshop papers because workshops have different (and generally lower) acceptance standards. In addition, the papers rejected from the main conference and invited to a workshop are labeled as rejects in our dataset. We extracted metadata and full text

[5] The definition of h5-index on Google Scholar Metrics: The h-index for articles published in the last 5 complete years. It is the largest number h such that h articles published in the past 5 years (2014-2018) have at least h citations each.

[6] https://scholar.google.es/citations?view_op=top_venues&hl=en&vq=eng_artificialintelligence

[7] https://openreview.net/group?id=ICLR.cc

from the newly collected papers with ScienceParse, which is an open source tool for parsing scientific papers (in PDF form). We split the parsed ICLR2018 and ICLR2019 data into train, validation, and test datasets, as Kang et al. did for ICLR 2017. The distribution is shown in Table 4.2. We will make the parsed data available as a benchmark dataset for further analysis.

4.4 Characterizing Accepted Papers

In this section, we will show the results of descriptive analysis and hypothesis testing for characterizing the difference between the accepted papers and rejected papers. In Section 4.4.1, we describe the statistical model, Welch's t-test (or unequal variance t-test) [391], we use to test the statistical difference between the accepted papers and the rejected papers. In Section 4.4.2, we examine the influence of the number of references in acceptance decisions. We further investigate whether citing recent papers would increase the chance of acceptance in Section 4.4.3. Then, we consider the relationship between the venues in which the referenced papers were published and the acceptance decision of the citing paper in Section 4.4.4. Lastly, in Section 4.4.5, we consider the h5-index of the references as a measure of impact to determine whether a higher average h5-index results in higher likelihood of acceptance. If not mentioned specifically, the analysis is conducted with the full 2,876 papers from ICLR from 2017 to 2019.

4.4.1 Welch's T-Test

To compare the central tendencies of two groups, statistically tests are natural choices. Student's t-test [361], Welch's t-test (or unequal variance t-test) [391], and Mann–Whitney U test [266] are commonly used to determine the probability of a given hypothesis is true. Researchers often assume normal distribution and homogeneous variance between their target groups. However, these conditions are commonly not satisfied [89] and Student's t-test can be strongly biased due to unequal variance. Recent studies [89, 330] have shown that Welch's test is a better candidate due to its better control of Type 1 error rates. Plus, we also observe heterogeneity of variance between the accepted papers and rejected paper in our dataset. Thus, we use Welch's t-test to conduct our analysis.

Welch's t-test and the degree of freedom ν is approximated with the Welch–Satterthwaite equation [336] as follows :

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \quad and \quad \nu \approx \frac{\left(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}\right)^2}{\frac{s_1^4}{N_1^2\nu_1} + \frac{s_2^4}{N_2^2\nu_2}} \tag{4.1}$$

where $\overline{X_i}$, s_i^2, and N_i are the sample mean, sample variance, sample size for group i, respectively, and $\nu_i = N_i - 1$, the degrees of freedom associated with the variance estimate for i group. After t and ν are computed, we can use these two statistics to verify each given hypothesis. We use significance level $\alpha = 0.001$ in all our analysis to suggest a robust interpretation of the null hypothesis.

4.4.2 Effect of the Number of References

First, we examine whether the number of references in a paper has any significant difference between the accepted papers and rejected papers. Several studies have used the number of reference as a feature to predict the impact of a paper [98, 429, 149, 419]. Dong et al. demonstrate that references are important, just behind the content and the venue of the paper, in predicting if a paper will increase its primary author's h-index. In this section, we use Welch's t-test to examine the following null hypothesis (H0):

H0: The mean of the number of references in rejected papers is equal to the accepted papers.

Fig. 4.1 is a box chart showing the mean and variance of number of references for overall ICLR papers and for each year. The consistent tendency that accepted papers have more references than rejected papers can be observed. We can also see that the number of references is increasing in both groups. We then conduct Welch's t-test to statistically verify if there is any significant difference on the number of references between the two groups. The results are shown in Table 4.3. Given the null hypothesis is true, the probability of the rejected papers have equal or larger number of references than accepted papers is less than the significance level ($\alpha = 0.1\%$). Thus, we can reject the null hypothesis in favor of the alternative hypothesis that the accepted papers have more references than the rejected papers.

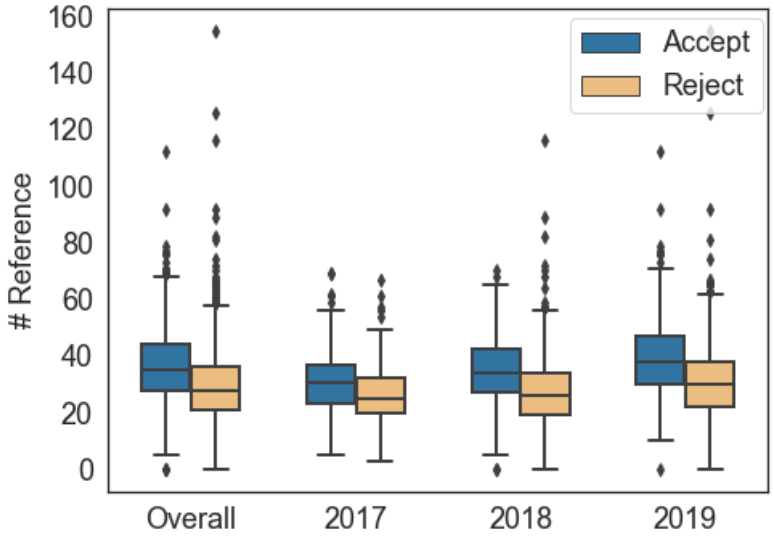


Figure 4.1: Box chart for number of references in accepted papers and rejected papers in ICLR over years. We observe a gradual increase in the number of references in both groups, but the accepted papers consistently have more references than rejected papers.

	Accept	**Reject**	**P Value**	**P<0.001**
Overall	36.422 ± 12.920	29.175 ± 12.839	9.78E-45	Yes
ICLR2017	31.198 ± 11.736	26.514 ± 10.294	2.95E-5	Yes
ICLR2018	34.744 ± 11.809	27.087 ± 12.326	8.34E-20	Yes
ICLR2019	39.335 ± 13.252	31.134 ± 13.396	8.49E-28	Yes

Table 4.3: The results of the Welch's t-test. Given the significance level $\alpha = 0.001$, we can safely reject our null hypothesis in favor of alternative hypothesis that accepted papers have more references than rejected papers in the ICLR dataset.

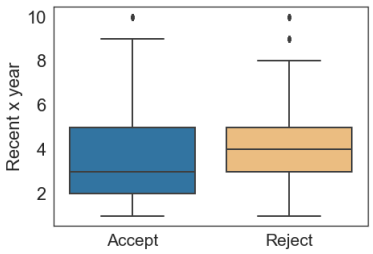


Figure 4.2: Year distribution of the aggregated publication years of all the papers in the two groups. Overall, accepted papers tend to refer to more recent publications than rejected papers.

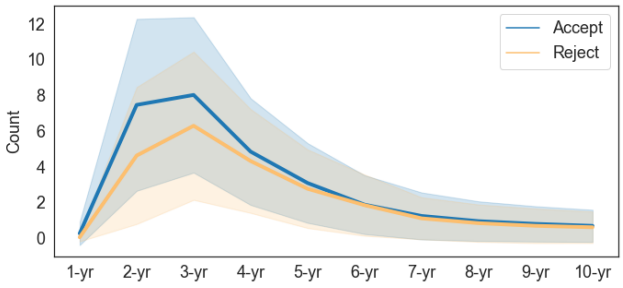


Figure 4.3: Reference distribution over the recent 10 years. The lines present the average of the number of references in every time distance and the shade indicates the variance. From the table, we can observe that accepted papers demonstrate greater amount of references to the publications in the recent 2 and 3 years. There is no much difference between the accepted paper and rejected paper after year distance = 5

4.4.3 Effects of Reference Recency

In this section, we consider the recency of references. We compute the difference between the publishing year of the paper and the publishing year of the reference to normalize the time difference, which we refer to as *year distance*. The year distance of a same-year reference is defined as 1. For example, a 2019 reference in a 2019 paper has distance 1 and a 2018 reference in a 2019 paper has distance 2. The reference counts in the most recent 10 years are shown in Figure 4.3. The figure shows the accepted papers demonstrate prominently higher number of references of recent 2, 3, and 4 year publications, while the average counts of references of over 5 years are similar between two groups. The box plot in Figure 4.2 delivers the same message that accepted papers overall cite more recent references than the rejected papers.

To statistically validate this observation, we formulate the following hypothesis:

H0: The mean of the year distance for rejected papers is equal to that of

accepted papers.

	Accept	**Reject**	**P Value**	**P<0.001**
Average year distance	3.956 ± 2.041	4.197 ± 2.048	1.99E-54	Yes
Proportion of 2nd year refs	0.208 ± 0.121	0.157 ± 0.120	6.07E-26	Yes
Proportion of 3rd year refs	0.218 ± 0.095	0.213 ± 0.115	0.1513	No
Proportion of 4th year refs	0.133 ± 0.071	0.146 ± 0.084	6.36E-6	Yes
Proportion of 5th year refs	0.084 ± 0.054	0.094 ± 0.070	1.51E-5	Yes

Table 4.4: Welch's t-test on reference year with a significance level of $\alpha = 0.001$. The accepted papers have a higher proportion of references at year distance 2 and smaller proportion of references at year distance 4 and 5.

Table 4.4 shows the results from hypothesis testing. Accepted papers have a lower value of time distance relative to rejected papers. At significance level $\alpha = 0.001$, we can reject the null hypothesis in favor of the alternative hypothesis that the accepted papers cite more recent publications. To further investigate the recency of the references, we break down the year distance and define the following hypotheses:

H0: There is no difference in mean proportion of references between the rejected papers and accepted papers at year distance=n.

To avoid the results being biased by the higher number of references in accepted papers, we normalize by calculating the proportion of the references for each year distance. Based on the results in table 4.4, we reject the null hypotheses for time distance at 2, 4, and 5. That is, there is significant difference between the accepted papers and rejected paper at time distance at 2, 4, and 5. We fail to reject the null hypothesis at time distance = 3. Moreover, we can observe the accepted papers have higher proportion of recent 2nd year references than rejected papers, while lower proportion of recent 4th and 5th year than rejected ones. These results are consistent with the recency hypothesis.

4.4.4 Effects of Reference Venue

	Accept	**Reject**	**Overall**
Num. of refs	36.422 ± 12.927	29.177 ± 12.846	31.722 ± 13.328
Num. of conf. refs	13.148 ± 7.454	10.280 ± 6.759	11.287 ± 7.142
Num. of Journals	2.513 ± 2.331	2.163 ± 2.144	2.286 ± 2.217
Num. of ArXiv	7.859 ± 6.546	6.620 ± 5.693	7.056 ± 6.034
Num. not indexed	12.114 ± 8.023	9.720 ± 7.051	10.561 ± 7.493

Table 4.5: The stats of the reference venues.

We then analyze the relationship between the venue of the references and the final decision of the paper. In this subsection, we examine the whether citing papers from top venues will increase the chance of getting accepted.

We compile our list of computer science venues and academic metrics from Guide2Research [8]. Guide2Research provides ranking of computer science conferences and journals based on their h5-index and impact factor, respectively. Guide2Research only includes computer science conferences and journals whose h5-index is greater than or equal to 12. In total, there are 424 computer science conferences and 650 computer science journals. We then lookup the venue for each reference in the compiled list. The basic venue statistics can be found in table 4.5. Of all the 91,201 references in our 2,876 ICLR papers, 60,838 references were successfully indexed. The remaining unindexed references include (1) those that have an abbreviated name format (e.g. ACM Trans. Graph.), (2) those from journals and conferences that do not appear in the Guide2Research list (e.g. Annual Review of Neuroscience), (3) those from online articles (e.g. `https://medium.com/ai-control/semi-supervised-reinforcement-learning-cf7d5375197f`), (4) those from PhD thesis (e.g. "PhD thesis, Deépartement d'Informatique et Recherche Opeérationnelle. Universiteé Montreéal" , and (5) those from non-English journals. Based on the statistics in Table

[8] `http://www.guide2research.com/`

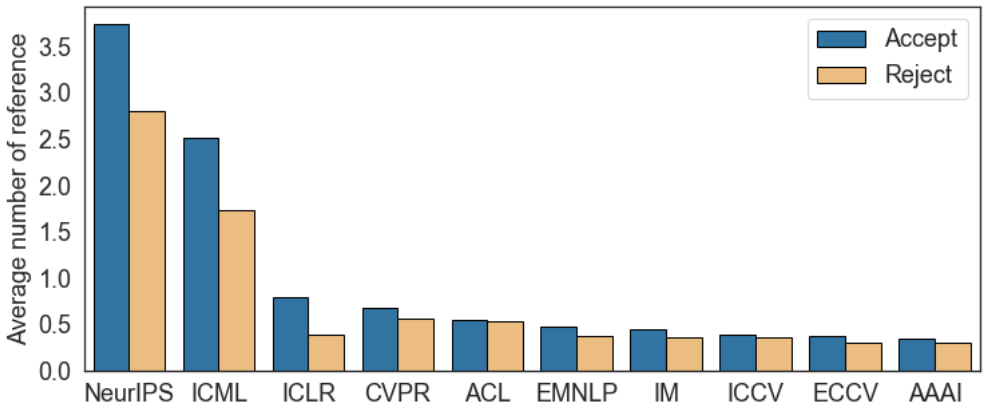


Figure 4.4: Reference distribution on the 10 most frequently referred top computer science venues. Overall, accepted papers refer to more top CS venues than rejected ones, especially to NeurIPS, ICML, and ICLR.

4.5, we focus on analyzing the references that are published through conferences due to the relatively low number of references from journals.

Figure 4.4 shows the reference distribution over the top 10 most frequent refer venues in ICLR datasets. The references are mostly from the top conferences in their disciplines. Based on the Google Scholar Metrics, NeurIPS, ICLR, ICML, and AAAI are the top 4 conferences in Artificial Intelligence. CVPR, ECCV, and ICCV are the top 3 conferences in Computer Vision and Pattern Recognition. ACL and EMNLP are the top 2 conferences in Computational Linguistics. We can also observe that accepted papers refer to significantly larger numbers of NeurIPS, ICML, and ICLR publications than the rejected papers.

To further study the influence of the venue of the references, we consider the following null hypothesis:

H0: There is no significant difference in the number of references from x conference between accepted papers and rejected papers.

Table 4.6 shows results for different venues. For NeurIPS, ICLR, and ICML (all in AI category), we reject the null hypothesis and support the alternative hypothesis that the

	Accept	**Reject**	**P value**	**P<0.001**
Artificial Intelligence				
NeurIPS	3.739 ± 2.683	2.804 ± 2.344	3.49E-20	Yes
ICLR	0.788 ± 1.894	0.393 ± 1.154	2.00E-9	Yes
ICML	2.510 ± 2.461	1.736 ± 1.916	9.98E-18	Yes
AAAI	0.3426 ± 0.789	0.311 ± 0.744	0.294	No
AISTATS	0.292 ± 0.611	0.205 ± 0.514	0.000127	Yes
Computer Vision and Pattern Recognition				
CVPR	0.678 ± 1.672	0.569 ± 1.362	0.076	No
ECCV	0.379 ± 0.915	0.302 ± 0.731	0.021	No
ICCV	0.389 ± 0.882	0.367 ± 0.824	0.514	No
Computational Linguistic				
ACL	0.5534 ± 1.652	0.53 ± 1.682	0.737	No
EMNLP	0.470 ± 1.341	0.379 ± 1.175	0.069	No

Table 4.6: Welch's t-Test on Reference venue. Accepted papers cite more references in artificial intelligence venues, while we can not conclude any significant differences in other disciplines.

accepted papers contain more references from these three conferences. For the rest of the conferences, we fail to reject the null hypothesis and cannot conclude that there is significant difference between the two groups.

4.4.5 The H5-index of References

In this section, we consider the influence of reference venue impact on acceptance. We associate each reference with the h5-index [158, 186] of its venue.

For each paper, we compute the total h5-index and the average h5-index:

$$\text{Total(h5-index)} = \sum_{j}^{N_r} h_j \qquad \textit{and} \qquad \text{ave(h5-index)} = \frac{\sum_{j}^{N_r} h_j}{N_r}$$

where h is the h5-index of the venue of each reference, and N_r is the total number of

references in a paper.

The visualization can be seen in Figure 4.5. We can observe rejected papers have relative lower h5-index than the accepted papers. However, the results might be biased by the fact that the rejected paper have overall lower number of references. We also see similar distributions between accepted papers and rejected papers in terms of average h5-index. We further investigate this problem statistically with following hypothesis:

H0: The mean h5-indexes of the venues for the rejected papers and accepted papers are equal.

We are also interested in knowing whether timing of the references have any influence in acceptance decision. Thus, we formulate the following hypothesis:

H0: There is no significant difference in terms of the average h5-index of the references' venues between accepted papers and rejected paper at time distance $= n$.

The hypothesis testing results can be found in Table 4.7. While the mean of the average h5-index in accepted papers is slightly higher than the rejected paper, we fail to reject (p value = 0.1110) the null hypothesis that there is no significant difference between accepted papers and rejected papers. However, if we look at the references at specific time distance, we are able to reject the null hypothesis and favor for alternative hypothesis that the mean of the average h5-index of accepted papers is significant higher than the rejected at time distance = 2, 4, and 5. For time distance over 5, there is no significant difference between the accepted papers and the rejected papers in terms of average h5-index of references' venues.

4.4.6 Distribution on arXiv.cl

We have shown the the accepted and rejected papers from ICLR exhibit statistically different distributions. However, ICLR may not be representative of other disciplines. We further analyze the arXiv.cl dataset compiled by Kang et al [191]. Kang et al. automatically labeled the papers in arxiv.cl (*Computation and Language*), arxiv.ai (*Artificial Intelligence*), and arxiv.lg (*Machine Learning*) within 2007-2017 as accepted or probably-rejected with respect to a group of top NLP, AI, and ML venues: ACL, EMNLP, NAACL, EACL, TACL, NIPS,

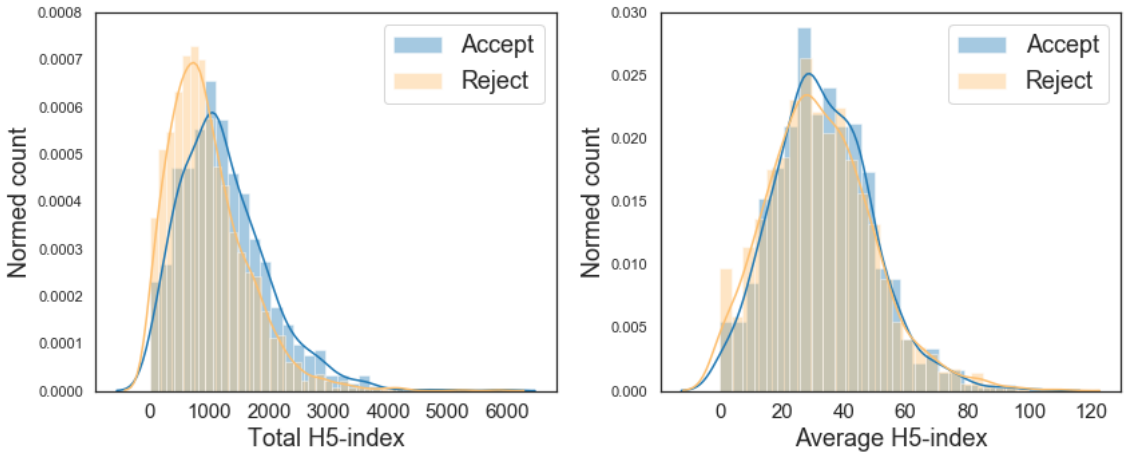


Figure 4.5: Normalized histogram of accepted/rejected papers over h-index. Accepted papers show a higher total h5-index. We can not observe a significant difference between accepted papers and rejected papers for the average h5-index.)

	Accept	**Reject**	**P value**	**P<0.001**
Ave. h5-index @ $D_{year} = 2$	22.914 ± 25.146	17.060 ± 26.407	5.61E-9	Yes
Ave. h5-index @ $D_{year} = 3$	40.795 ± 29.807	37.421 ± 31.603	0.00462	No
Ave. h5-index @ $D_{year} = 4$	43.139 ± 34.744	38.313 ± 35.629	4.37E-4	Yes
Ave. h5-index @ $D_{year} = 5$	40.731 ± 38.643	35.328 ± 39.904	4.15E-4	Yes
Ave. h5-index @ $D_{year} > 5$	29.540 ± 21.360	30.508 ± 23.425	0.262	No
Ave. h5-index	33.424 ± 15.915	32.421 ± 17.252	0.1110	No

Table 4.7: Welch's t-Test on h5-index with a significant level of $\alpha = 0.001$. There is no significant difference between two groups on average h5-index, while accepted papers have higher average h5-index at time distance = 3 than rejected ones.

ICML, ICLR and AAAI. The accepted papers are assigned when the titles and the authors' name of the arXiv submissions match an accepted paper in the target venue. Because

the target venues are mostly NLP-related, we consider the number of the references and the average year distance of the references in the NLP category. Figure 4.6 shows that the results for submissions in *Computation and Language* are similar to those for the ICLR dataset. Accepted papers consistently cite more papers and more recent papers over multiple years. Since the labels are derived from PeerRead rather than ground truth, we consider these results as an auxiliary validation only.

4.5 Predicting Acceptance

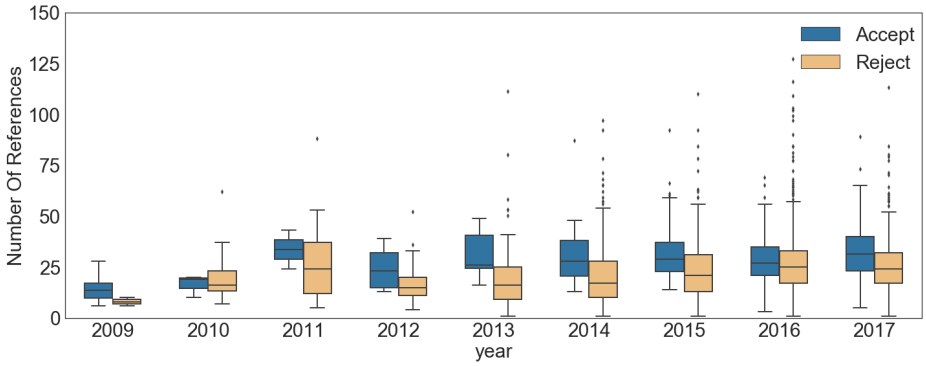
In this section, we investigate bibliographic features as predictors of acceptance. We find that a model trained with reference features alone is comparable with the state-of-the-art NLP based models. We will describe the competitive methods in Section 4.5.1, followed by the evaluation metrics we use in Section 4.5.2. We describe implementation details in Section 4.5.3 and report the results in Section 4.5.4.

4.5.1 Competitive Methods

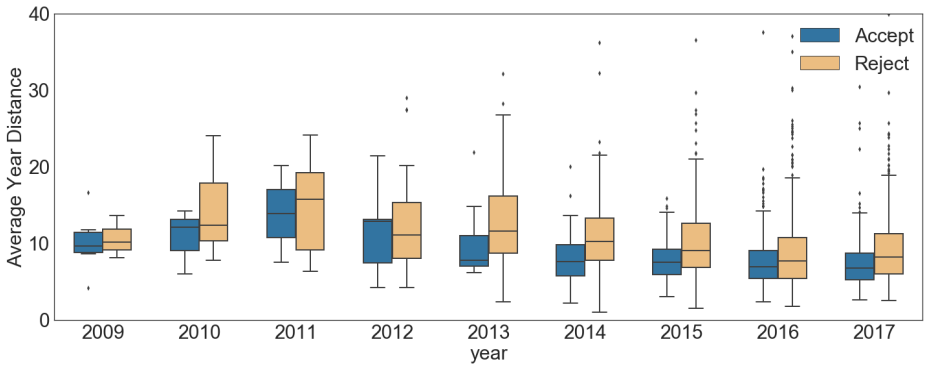
We compare our model against two state-of-the-art models:

- **PeerRead [191]:** Kang et al. organized and annotated papers and reviews from multiple computer science conferences. They also extracted metadata and full text from the papers. They further provided a baseline for the acceptance prediction task with a hybrid of content-based and structural-based features.

- **DeepSentiPeer [130]:** Ghosal et al. tackled the automated peer review problem by integrating review sentiment. To do this, the authors embedded the published papers and the reviews with the Transformer variant of the Universal Sentence Encoder (USE) [57] and concatenated the sentence embeddings to represent the documents. The review sentiment encoding is acquired using the concatenation of the sentence encodings, the Valence Aware Dictionary and sEntiment Reasoner (VADER) [175]. The authors then utilized Convolutional Neural Networks to capture the features from the paper and review representations. Finally, the decision is obtained using a Multi-layer



(a)



(b)

Figure 4.6: (a) The distribution of the number of references between the accepted papers and the "probably rejected" papers on *arxiv.cl* over year. (b) The distribution of the averaged year distance of the references between the accepted papers and the "probably rejected" papers on *arxiv.cl* over year. We can observe that the accepted papers refer to more recent publications and have more references in their submissions. This finding is consistent with the ICLR datasets.

Perceptron with a joint vector of the paper representation, the review representation, and the sentiment encoding.

4.5.2 Evaluation Metrics

In both PeerRead and DeepSentiPeer, the authors assessed the model performance of the decision prediction task with only one metric, accuracy, which is defined as:

$$acc = \frac{TP}{N_p} \tag{4.2}$$

where TP means true positive, the correct predicted papers, and N_p is the total number of papers in the test set. Even though the accuracy metric can reflect overall performance of a model, the evaluation is not quite comprehensive and sometimes misleading, especially when we have imbalanced classes. We are proposing to include precision and recall in the evaluation metrics for paper acceptance prediction. Precision can be seen as a measure of quality, while recall can reflect the completeness. The precision and recall are defined as:

$$Precision = \frac{TP}{TP + FP} \qquad and \qquad Recall = \frac{TP}{TP + FN}$$

where FP means false positives, the rejected papers are wrongly predicted as accept, and FN means false negatives, the accepted papers are incorrectly predicted as reject. Low precision indicates the model misses some high quality papers and low recall means the model includes papers with lower quality. We can evaluate the models with more perspective with more metrics available.

4.5.3 Implementation

We use 16 coarse bibliographic features that we found can discern acceptance. The list can be seen in the Appendix. Because some of our features are highly correlated, we use Principal Component Analysis (PCA) to address multicollinearity [114, 267] issues, following Lafil et al. [225]. We use PCA to transform the features into 5 components and use the components as features to predict acceptance. We evaluate with six different machine learning models: Logistic Regression, Support Vector Machine (SVM) with RBF kernal, Multi-layer Perception, AdaBosst, and Random Forest. We train our models with the training set and use

the validation set to determine regularization settings, then report performance on the test set. We also consider imbalanced classes in our training set. We train on all available data rather than artificially balancing the classes, a transformation that has been shown to have limited (if any) benefit [83, 196], particularly for logistic regression. For PeerRead [191] and DeepSentiPeer[130], we use their code [9] and follow their instructions and hyperparameter settings described in the paper to train their models. Sometimes, models tend to predict an unrealistically low number of accepted papers. In these cases, we adjust the threshold so the model predicts around 30% of papers as accepted, which is roughly the acceptance rate for ICLR.

4.5.4 Results

We consider the following questions: (1) Are the bibliographic features predictive for paper acceptance, relative to the metadata and learned text features used by NLP models? (2) Does performance improve with the availability of more data? (3) Can existing models be improved by including bibliographic features? (4) Does the model still perform well if we ignore the current year, which is a more realistic scenario in practice? (5) Is a minimal model using only the most predictive bibliographic features still effective?

Acceptance Prediction In an Individual Year We conduct analysis on ICLR2017, ICLR2018, and ICLR2019 to assess the ability of references features to predict acceptance decision. We also run experiments with PeerRead and DeepSentiPeer as comparison. The empirical results are shown in Figure 4.7. We call models trained with bibliographic features as *BibOnly*. DeepSentiPeer also considers review embedding and review sentiment to predict acceptance decision, while PeerRead and BibOnly only extract features from the papers. Except for ICLR2017, BibOnly outperforms PeerRead and DeepSentiPeer. We also observe that BibOnly has higher precision than other models. We are surprised to see the inferior performance from DeepSentiPeer. With more information available for DeepSentiPeer, DeepSentiPeer has natural advantages over PeerRead and reference features. Our

[9] PeerRead: https://github.com/allenai/PeerRead, DeepSentiPeer: https://github.com/aritzzz/DeepSentiPeer

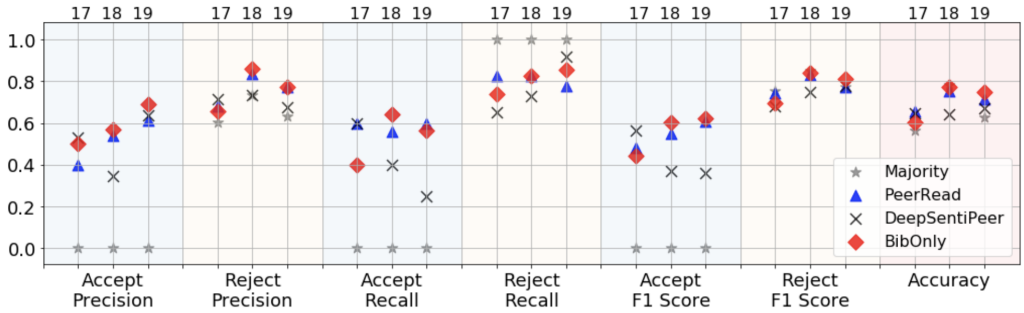


Figure 4.7: Experimental results for predicting paper acceptance on ICLR2017, ICLR2018, and ICLR2019. DeepSentiPeer includes review embeddings and review sentiment in their models, whereas PeerRead and BibOnly only look at the features from the papers. Except for ICLR2017, reference features outperform PeerRead and DeepSentiPeer based on accuracy and F1-Score. The poor performance on ICLR2017 might be due to the relative low data in ICLR2017.

interpretation of the poor performance from DeepSentiPeer is insufficient data for training.

Model	Train Year	Test Year	Precision		Recall		F1-Score		Accuracy	Δ*
			Accept	Reject	Accept	Reject	Accept	Reject		
PeerRead	17 & 18	18	0.552	0.864	0.640	0.814	0.593	0.838	0.768	+1.5%
DeepSentiPeer			0.464	0.821	0.520	0.786	0.491	0.803	0.746	+10.5%
BibOnly			0.607	0.877	0.680	0.838	0.642	0.857	0.796	+2.2%
PeerRead	17, 18, 19	19	0.667	0.769	0.571	0.833	0.615	0.800	0.737	+2.4%
DeepSentiPeer			0.865	0.890	0.804	0.927	0.833	0.908	0.881	+20.8%
BibOnly			0.674	0.769	0.564	0.842	0.614	0.804	0.740	-0.7%

Table 4.8: Results on training with more data. *Δ indicates the accuracy difference from the corresponding model and year of the test set in Figure 4.7. We can observe the improvement for all models, especially DeepSentiPeer. DeepSentiPeer increases 10.5% and 20.8% for 2018 and 2019, respectively.

Model	Train Year	Test Year	Precision		Recall		F1-Score		Accuracy	Δ*
			Accept	Δ*	Accept	Δ*	Accept	Δ*		
PeerRead+Bib	17, 18	18	0.615	+6.3%	0.640	+0.0%	0.627	+3.4%	0.796	+2.8%
PeerRead+Bib	17, 18, 19	19	0.660	-0.7%	0.636	+6.4%	0.648	+3.3%	0.747	+1.0%
DeepSentiPeer+Bib	17, 18	18	0.538	+7.4%	0.840	+32.0%	0.656	+16.5%	0.769	+2.3%
DeepSentiPeer+Bib	17, 18, 19	19	0.868	+0.3%	0.821	+1.7%	0.844	+1.1%	0.892	+1.1%

Table 4.9: Experimental results on combining baseline model PeerRead and NLP model DeepSentiPeer with our reference features (Bib). $\Delta*$ indicates the result difference from the corresponding model and dataset. We can see that adding reference features can consistently improve the performance of other models in almost every metric.

Is More Data Beneficial? In this experiment, we analyze the benefits of more data in acceptance prediction task. In general knowledge, more data can significantly improve the model performance in machine learning and deep learning tasks. However, time sensitivity matters in acceptance prediction. For example, an accepted paper now might not be accepted in five years because of fast advance of computer science research. Therefore, we investigate whether or not more data is beneficial in paper acceptance prediction. Ghosal et al. [130] trained their sentiment-based model with ICLR2017's and ICLR2018's papers and reviews to increase the amount of the training data and evaluated the model on ICLR2017 papers. However, in reality, we do not have access to the future papers and the reviews. The model should not be fed with the future papers and reviews for training, while the papers from the past are available for use. In our experiments, we train the models with papers from the 2017 and 2018 training sets, and evaluate on the 2018 test set. We also experiment on training with papers in training sets from 2017 to 2019 and test on the 2019 test set. The experimental results are shown in Table 4.8. We can observe that consistent improvement are achieved on all models and reference features still demonstrate superior performance than PeerRead features with more data available. The performance of DeepSentiPeer has improved by a large margin with more data available for training. This is understandable because DeepSentiPeer is based on deep neural networks which require large data size to

perform well.

Can Bibliographic Features Improve Existing Models? In this experiment, we investigate whether including bibliographic features improve the performance of NLP models. For PeerRead, we concatenate our reference features with their coarse features as input for the machine learning models. For DeepSentiPeer, we concatenate our reference features with the feature vector at the Feature-Level Fusion step before the Multi-Layer Perception prediction. Table 4.9 demonstrate the outcome of this experiment. Reference features improve PeerRead and DeepSentiPeer in both datasets for almost all the metrics.

Predicting Acceptance from Previous Data We simulate a scenario where a model for 2019 is trained with the papers from 2017 and 2018 only. We train all three models with the training set from the ICLR2017 and ICLR 2018, validate on the validation set from the ICLR2017 and ICLR2018, and evaluate on the the ICLR2019 dataset. We also assess the competitive models with the bibliographic features included. The results are shown in Table 4.10. Bibliographic features show superior results on their own and significantly improve the competitive models. Our interpretation of this outcome is that reference usage is consistent across years and that reference features are not time sensitive. On the other hand, same content of an accepted paper might not be accepted in five years. The time sensitivity of the acceptance prediction is the challenge for NLP-based models.

A Simpler Predictive Model Principal components are hard to interpret. We experiment with combinations with 2 and 3 features to find the most predictive reference features. The experimental results are shown in Table 4.11. The consistent pattern we can observe from both datasets is that same-venue referencing (The number of refereces from ICLR) and the recency of the references (The number of the references with year distance = 2 and The Averaged H5-Index of The Reference Venues with Year Distance = 2) are very predictive in predicting acceptance. With only two reference features, the model accuracy is only 0.3% shy from the PeerRead baseline and the model outperforms DeepSentiPeer in one of the combined datasets.

Model	Precision		Recall		F1-Score		Accuracy
	Accept	Reject	Accept	Reject	Accept	Reject	
PeerRead	0.537	0.794	0.631	0.724	0.580	0.758	0.693
DeepSentiPeer	0.389	0.813	0.560	0.685	0.459	0.744	0.652
BibOnly	0.629	0.753	0.437	0.869	0.516	0.807	0.724
PeerRead+Bib	0.599	0.788	0.573	0.805	0.586	0.797	0.727
DeepSentiPeer+Bib	0.533	0.862	0.640	0.800	0.582	0.830	0.758

Table 4.10: We conduct experiments that designed to be close to the real life setting, where we train the models on the older papers (ICLR2017 and ICLR2018) and evaluate on the latest papers (ICLR2019). Our reference features outperforms the baseline PeerRead model and DeepSentiPeer by a noticeable margin. In addition, adding the reference features to the two competitive models improves the performance significantly.

4.6 Discussion

In this study, we investigate the relationship between references — in terms of the quantity, recency, impact, and discipline similarity — and acceptance decisions. We find that accepted papers cite more publications, and specifically, more recent ones. Santini et al. [335] argue that the sufficient number of references can reflect authors' attention to details and their efforts of extensive literature reviews. Especially in computer science community, it is not uncommon to see best-performing models be overtaken within months of publication. Thus, reviewers value the recency and completeness of the references on a topic. Several studies also use the impact of references to predict the impact of the citing paper. Dong et all compute the ratio of references that have at least a certain value of h-index to the total references and the average number of citations accumulated by all the references as the "reference feature" of a paper [98]. They find that reference features are ranked 6th and 7th out of 26 features in terms of importance in predicting impact. The method to quantify the quality of references in Dong et al.'s work is different from our method, but it delivers

Train Year	Test Year	Features	Accuracy
17, 18	18	Number of The References from ICLR Number of The References with Year Distance = 2	0.763
17, 18	18	Number of The References from ICLR Number of The References with Year Distance = 2 Percentage of The References with Year Distance = 4	0.763
17, 18, 19	19	Number of References Averaged H5-Index of The Reference Venues with Year Distance = 2	0.693
17, 18, 19	19	Number of The References Number of The References from ICLR Averaged H5-Index of The Reference Venues with Year Distance = 2	0.707

Table 4.11: Experiment on the most predictive reference features. The consistent pattern between the two datasets is that same-venue referencing and the recency of the references are very essential in predicting paper acceptance.

a similar message: it is important to "stand on the shoulder of giants" [82].

It is surprising to see a model trained with simple reference features outperform a more complicated NLP model, which considers entire reviews and review sentiment. It is not the first time that the structural information has shown more promising application than content. Just as PageRank [304] showed that the hyperlink structure of the web determined the importance of a webpage (more so than even its content), the citation connections between scientific publications can be used to predict the quality of the papers. The findings in this study open up a new direction based on reference analysis and its use in applying AI to the peer review process. Can a citation network representation learning model improve upon the performance? Also, even though the promising results presented in this study, there are limitations of these reference features in terms of automated peer review process. Good reference usage can be a latent feature of high quality papers, but the quality of a submission should still of course be based on its content and the contribution to the community. Models relying on bibliographic information may also lose effectiveness over time, as the community

adapts to change their referencing habits based on the findings in this study, eliminating the divergence between accepts and rejects.

Our findings also raise the question of why references have such predictive power. Do high quality papers share similar patterns of reference usage or do the reviewers over-reliant on references as a proxy for studying the content of the paper? What do these results say about the limitations of NLP-approaches for predicting acceptance? While referencing recent publications from top conferences demonstrates sufficient awareness and knowledge of a subject, these patterns may also suggest a degree of community insularity that inhibits communication across sub-disciplines, and potentially a "short memory" for prior work. The recency of references may also be an artifact of the current empirical trend in machine learning, characterized by rapid and incremental improvement of the state-of-the-art.

4.7 Summary

In this study, we present the results indicating that accepted papers from ICLR cite more papers, more recent papers, more papers from similar disciplinary venues, more recent papers from high impact venues. We also show that models trained with these bibliographic features alone outperforms NLP-based models, one of which even take reviews and the sentiment of the reviews into account. We further demonstrate that the combined reference features and NLP-based models can produce even better performance. Finally, we present a model trained with just two reference features, the number of reference within the last two years and the number of references from the same venue (ICLR), can achieve accuracy of within 3% of the accuracy of the best known model. The findings from this study provide a new direction for incorporating AI into peer review process and a different perspective into peer review itself.

4.8 Supplementary Materials

The venue abbreviations and full names mentioned in this study: **AAAI**: AAAI Conference on Artificial Intelligence, **ACL**: Meeting of the Association for Computational Linguistics, **AISTATS**: International Conference on Artificial Intelligence and Statistics, **CIKM**: ACM International Conference on Information and Knowledge Management, **CoNLL**: Confer-

ence on Computational Natural Language Learning, **CVPR**: IEEE/CVF Conference on Computer Vision and Pattern Recognition, **ECCV**: European Conference on Computer Vision, **EMNLP**: Conference on Empirical Methods in Natural Language Processing, **ICCV**: IEEE/CVF International Conference on Computer Vision, **ICLR**: International Conference on Learning Representations, **ICML**: International Conference on Machine Learning, **IM**: IFIP/IEEE International Symposium on Integrated Network Management, **NeurIPS**: Conference on Neural Information Processing Systems.

Table 4.12 is the list of the reference features:

Reference Feature List
The number of references
The averaged year distance of the references
The percentage of the references with year distance = 2
The percentage of the references with year distance = 4
The percentage of the references with year distance = 5
The percentage of the references with year distance > 5
The number of references from NeurIPS
The number of references from ICLR
The number of references from ICML
The number of references from AISTATS
The averaged H5-index of the references venue
The averaged H5-index of the references venue with year distance = 2
The averaged H5-index of the references venue with year distance = 3
The first principal component of the number of the references from top conferences
The number of the references with year distance $\leqslant 1$
The number of the references with year distance $\leqslant 2$

Table 4.12: Bibliographic features for predicting acceptance.

Chapter 5

LEVERAGING APPLICATION-AGNOSTIC ONTOLOGIES WITH HIERARCHICAL MULTI-LABEL CLASSIFICATION

5.1 Introduction

Hierarchy- and graph-structured domains are becoming ubiquitous for learning tasks on and off the web. In high-expertise domains, human attention tends to be invested in designing and curating application-agnostic ontologies rather than on hand-labeling application-specific datasets. WordNet, first introduced by Miller et al. [279] in 1998 and since adopted in over 200 languages and used in tens of thousands of papers, is characterized by rich thesaurus relationships between terms. ImageNet [91], designed as a WordNet analoogue for images, contains over 3 million images labeled with 60,942 terms organized into a hierarchy (actually a DAG, but represented as a hierarchy by repeating nodes). For learning tasks on the web, ontological relationships between labels have been used to improve document classification [298, 283], study zero-shot learning [128], and improve recommendation systems [100, 432, 333, 219].

Despite the opportunity for supervision, relationships among labels are often ignored. For example, the ImageNet dataset led to AlexNet [220], ResNet [151], and VGG [347], but all three models ignore the hierarchical structure of the labels. In response, an emerging community is studying hierarchical multi-label classification, aiming to use the hierarchy to help supervise multi-label learning.

Current hierarchical multi-label classification models, however, tend to emphasize particular domains and are computationally expensive to train. Wehrmann et al. [390] proposed HMCN, a cascade neural network that simultaneously optimizes local hierarchical label relationships and the global hierarchy while penalizing hierarchical violations. However, the amount of the parameter of HMCN-F grows with the number of hierarchical levels. It gets computationally expensive when the hierarchies are large. Xu et al. [407] introduced a

hierarchical classification model that represents the correlation among labels with the label distribution and learns a mapping function from the instance to the label distribution, but the model is only tested for single-label problem. HyperIM [67] learns label-aware document representations and model the word and label hierarchies in hyperbolic space, but the model is only suitable for the text domain.

Existing approaches to hierarchical multi-label classification also tend to assume the availability of data to provide balanced coverage of the ontology, a condition that becomes increasingly unlikely as the ontology grows in size, and becomes impossible when the ontology is bigger than the training data or when the ontology changes after training data has been collected. This latter situation is especially insidious: ontologies undergo constant revision in practice, meaning that training datasets face continual obsolescence.

While several methods aim to prevent hierarchy violations (recommending a child label without also recommending a parent label), there has been no principled model-agnostic evaluations of hierarchy violations. Despite the emphasis on preventing hierarchy violations by Wehrmann et al. [390] and Giunchiglia et al[131], no direct evaluation of hierarchy violations was conducted.

In this study, we propose an algorithm for hierarchical multi-label classification that separates the problem into learning reusable embeddings of the ontology itself, then training a classifier using these embeddings. Since the structure of the ontology is encoded in the learned representation, trained classifiers can predict a correct label even with no examples of that label available, which in turn allows a trained classifier to be more robust when the ontology changes. We use a graph autoencoder to learn label representations from the structure of the ontology, then use a simple model to learn the function mapping the input space into the label embedding space using binary cross entropy loss. We call this framework Surj .

We show that Surj outperforms current state-of-the-art algorithms on 20 benchmark datasets, usually by a significant margin. We also propose one metric to quantify hierarchy violations, Global Hierarchy Violation. The metric is applied to Surj and it reveals that there is no hierarchy violation from Surj in all 20 datasets. Surj is also trained up to 40 times faster than the current state-of-the-art C-HMCNN. We demonstrate the effectiveness

of the ontology learning with an ablation study. We also show that Surj is robust to lower data size and remain superior even when only half of the training data is provided.

5.2 Related Work

Hierarchical multi-label classification has been a long standing problem due to the ubiquity of ontologies. Ontologies are used to organize knowledge in a wide variety of domains on the web [355, 354], in urban settings [4, 112], finance [358, 368], oceanography [333], and art [86]. Some models are tailored to Natural Language Processing for text multi-label classification [65, 67, 257, 172]. While these models present novel networks, they are limited to NLP applications and not applicable in our evaluation. We review studies for general hierarchical multi-label classification problem.

Hierarchical Multi-label Classification Most HMC algorithms can be categorized into global and local approaches. Global approaches are designed to handle the entire hierarchy. Vens et al. proposed Clus-HMC [378], based on the concept of Predictive Clustering Trees and involving a decision tree learner to map the entire hierarchy. Schietgat et al. extended the idea from Clus-HMC in Clus-Ens [338], adopting a bagging strategy to create decision tree ensembles. MHC-CNN [44] consists of a Competitive Neural Network where each neuron represents a node in the hierarchy and the whole network is a clone of the hierarchy. Masera et al. [271] proposes AWX (Adjacency Wrapping Matrix), which incorporates the hierarchy into their model architecture and the learned knowledge propagates through each layer. C-HMCNN [131] has a hierarchy-coherent layer to produce coherent predictions by construction.

Local approaches break down the problem into smaller classification tasks, often by level [58, 59, 60, 247, 431] or by node[62, 118, 36, 223, 407]. Cerri et al. [58, 59, 60] proposed HMC-LMLP, a approach based on a chain of Multi-Layer Perceptrons (MLPs) with a single layer represents a level in the hierarchy. The input of a given MLP is the output of the previous MLP. Bi et al.[36] utilize Kernel Dependency Estimation (KDE) to transforms the number of labels in a hierarchy into a workable number of single-label learning problems. Condensing Sort and Select Algorithm (CSSA) is employed to find an optimal approximation subtree to

preserve tree structure and they use ridge regression in the learning step. Wehrmann et al. [390] presented HMCN-R and HMCN-F, which are optimized by a global and a local loss. HMCN-F is a feed-forward network and HMCN-R is a recurrent architecture. While HMCN-F produces better overall results but the network parameters increase significantly as the hierarchy grows. While there has been significant progress on this problem, our approach of directly embedding the ontology using graph neural networks has not been considered, and as we will show, offers significant improvements. We explore how graph neural networks can contribute to hierarchical multi-label classification and review graph representation methods in the next subsection.

Deep Learning on Graphs Graph representation learning with graph neural networks has created opportunities for applications such as node classification, link prediction, and spatial–temporal graph forecasting. Jurisch et al. [189] utilizes graph convolution networks to solve ontology alignment problem. Schlichtkrull et al. [339] models relational data with graph convolutional networks. Zang et al. [426] apply graph embeddings for gene ontology annotations to predict protein-protein interaction. In our work, we propose to learn node embeddings to capture the structure information of a hierarchy and use these embeddings to train a multi-label classifier.

5.3 Method

In this section, we define the problem of hierarchical multi-label classification and introduce our framework.

5.3.1 Preliminaries

We define a label hierarchy $H = (V, E)$ as a graph with labels V and edges E, where edges are typically parent-child relationships representing specificity. We are given a space of input instances X where each element is a vector of features $x_1, x_2, ..., x_N$.

Our framework consists of a learned function $g : V \rightarrow Z^D$ that embeds each vertex in the ontology into a D-dimensional representation, and another learned function $m : X \rightarrow P(R)$ to take input instances and produces a probability distribution over the learned embedding

space R. The classifier is trained to map the input instances X into an embedding space R , followed by a cosine similarity layer to compute the cosine similarity between the embedding space and the ontology representation. The classifier is optimized with the binary cross entropy (BCE). The intuition behind this framework is that graph representation learning models allow us to capture the structural information from the label hierarchies: labels that are "close" in the hierarchy are "close" in the learned space. The learned embedding allows us to accommodate complex graphs, enforce parent-child relationships in predicted labels, make better use of limited training data, and tolerate ontology changes without requiring new training data.

5.3.2 Ontology Learning

For the label hierarchy learning, we use a graph auto-encoder [197] to learn the node embeddings. Graph auto-encoders are easy to implement and computationally efficient. We introduce an adjacency matrix A constructed from the label graph H and its degree matrix D. Node features are constructed in an $N \times D$ matrix $\mathbf{S}$. The graph autoencoder includes a two-layer Graph Convolutional Network (GCN) defined as:

$$\hat{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^T), \text{ with } \mathbf{Z} = GCN(\mathbf{S}, \mathbf{A}) \tag{5.1}$$

The two-layer GCN is defined as:

$$\text{GCN}(\mathbf{S}, \mathbf{A}) = \hat{\mathbf{A}}\text{ReLU}(\hat{\mathbf{A}}\mathbf{S}\mathbf{W_0})\mathbf{W_1} \tag{5.2}$$

where $\text{ReLU}(\cdots) = max(0, \cdots)$, $\mathbf{W_i}$ suggests weights, and $\hat{\mathbf{A}} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized adjacency matrix. Finally, $\sigma(\cdots)$ is the logistic sigmoid function.

Hierarchy nodes are frequently equipped with semantic features. For example, every node of WordNet consists of words which can be converted into word embeddings. Protein and gene possesses observed features within the hierarchies. While there are no feature vectors provided for the node in the benchmark dataset, the learned embeddings with feature vectors provided can be more representative and further improve the overall HMC performance. The graph autoencoder takes both dense feature vectors and an affinity matrix as inputs, allowing the ontology learner to learn the semantic and latent features of the nodes.

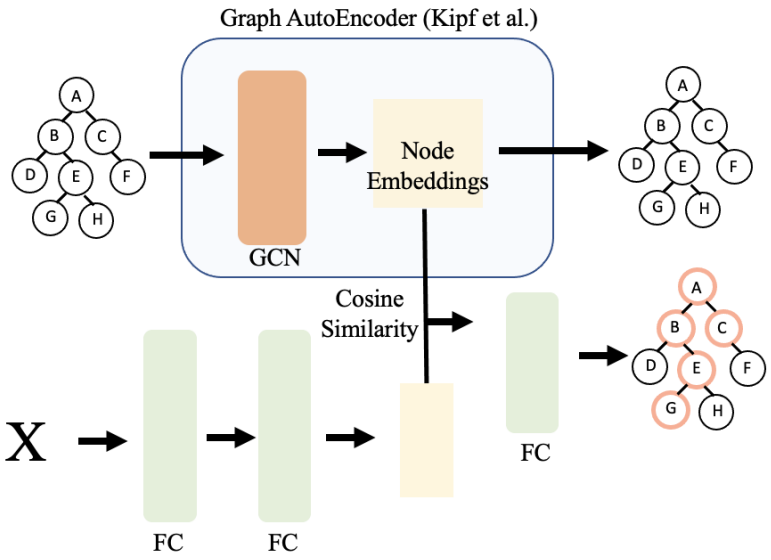


Figure 5.1: Illustration of our framework. We learn a representation for the label ontology using a graph autoencoder. Then, the model considers the node embeddings and maps the input instances X onto the node embedding space with cosine similarity. Finally, the model is optimized with binary cross entropy and produce probability confidence as output.

We perform an experiment to verify this premise using the ontology from one of the benchmark datasets, cellcycle(FUN).

For this framework to succeed, the learned hierarchy embedding needs to reflect the structural information. To show a simple verification of the correspondence, we take the label hierarchy from the CellCycle(FUN) dataset and compare the shortest path of all node pairs and the cosine similarity of their learned embeddings from the graph autoencoder. The results are shown in Figure 5.2. The x-axis is the length of the shortest path and the y-axis is the cosine distance d_{cosine}, defined as:

$$d_{cosine} = 1 - cos(\boldsymbol{P}, \boldsymbol{Q}) \tag{5.3}$$

where

$$cos(\boldsymbol{P}, \boldsymbol{Q}) = \frac{\boldsymbol{P} \cdot \boldsymbol{Q}}{||\boldsymbol{P}|| \cdot ||\boldsymbol{Q}||} \tag{5.4}$$

$\boldsymbol{P}$ and $\boldsymbol{Q}$ are two dense vectors. We can simply observe from the figure that the closer the two nodes are in the ontology, the more similar they are in the learned embedding space.

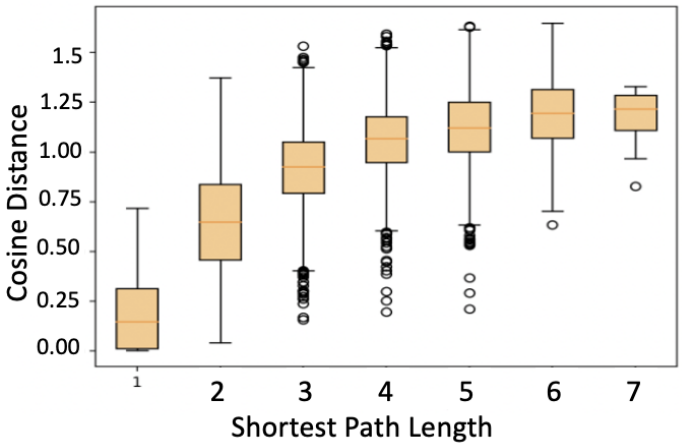


Figure 5.2: The relationships between the shortest path length of all node pairs and the cosine similarity of their learned embeddings. The closer the two nodes are in the tree, the more similar they are in the embedding space.

5.3.3 Multi-label Classification

The framework leverages the learned representation by learning to map the input instances on to the learned node representations. This is achieved by inserting a cosine similarity layer to compute the cosine similarity between the output from the fully connected layers and the node embeddings. The cosine similarity layer follows equation 5.4.

The cosine similarity layer is followed by another fully connected layer and a sigmoid layer for multi-label classification. The model is optimized by Binary Cross Entropy Loss:

$$L = -\frac{1}{V}\sum_{i=1}^{V} y_i \times \log(p(y_i) + (1 - y_i)) \times \log(1 - (p(y_i))) \tag{5.5}$$

where $y \in 0, 1$ is the label and $p(y)$ is the predicted probability of the node being true.

Table 5.1: Datasets used in the evaluation

Taxonomy	Dataset	# Classes	# Attributes	Depth	Train	Validation	Test
Tree	ENRON	56	1000	3	692	296	660
	DIATOMS	398	371	3	1085	464	1054
	IMCLEF07A	96	80	3	7000	3000	1006
	IMCLEF07D	46	80	3	2100	900	3000
FUNCAT (Tree)	CELLCYCLE	499	77	6	1628	848	1281
	DERSI	499	63	6	1608	842	1275
	EISEN	461	79	6	1058	529	837
	EXPR	499	551	6	1638	849	1291
	GASCH1	499	173	6	1634	846	1284
	GASCH2	499	52	6	1639	849	1291
	SEQ	499	478	6	1701	879	1339
	SPO	499	80	6	1600	837	1266
Gene Ontology	CELLCYCLE	4122	77	12	1625	848	1278
	DERSI	4116	63	12	1605	842	1272
	EISEN	3570	79	12	1055	528	835
	EXPR	4128	551	12	1636	849	1288
	GASCH1	4122	173	12	1631	846	1281
	GASCH2	4128	52	12	1636	849	1288
	SEQ	4130	478	12	1692	876	1332
	SPO	4116	80	12	1597	837	1263

5.4 Experimental Setup

In this section, we describe the empirical experimentation to verify the effectiveness of Surj for hierarchical multi-label classification. We evaluate Surj on 20 benchmark datasets across biological sequencing (protein function prediction), images, and text and compare our

framework against six other state-of-the-art algorithms in the HMC space. While several recent HMC models are designed based on the premise of avoiding hierarchy violations, there arre no metrics to quantitatively measure hierarchy violations. Global Hierarchy Violations proposed to determine the magnitude of the hierarchy violations of the predicted outputs. The implementation of the experiments are detailed in Sec. 5.4.4.

5.4.1 Datasets

We consider 20 datasets across multiple domains used in previous hierarchical multi-label classification studies [390, 131]. The datasets consist of protein function prediction [70], annotation of medical images [95, 96], or text classification [201]. The datasets are constructed as trees (MIPS functional Catalogue for protein function) or directed acyclic graphs (Gene Ontology). The statistics for the datasets are shown in Table 5.1.

As mentioned by Wehrmann et al [390], these datasets are challenging for neural networks for multiple reasons: (1) The training samples are relatively low. (2) The number of features vary significantly across the datasets (varying from 52 to 1000) (3) The hierarchies exhibit a wide range of depths and number of classes.

5.4.2 Competitive Methods

We compare Surj against several models that are considered the state-of-the-art for Multi-label Hierarchical Classification:

- C-HMCNN [131]: C-HMCNN leverages a constraint layer to ensure the predictions are coherent with the hierarchy constraints.

- HMCN [390]: Wehrmann et al. two neural network architectures, HMCN-F and HMCN-R, for HMC based on the concept of finding the local hierarhical class-relationships and entire class hierarchy with penalizing hierarchical violations. HMCN-F is a feed forward network designed for optimizing the hierarhical structure of the labeled data and HMCN-R is a recurrent network where the global flow shares weights throughout the hierarchy.

- HMC-LMLP [60]: HMC-LMLP is the first study to utilize neural networks for HMC problems. They associate one multi-layer perception (MLP) to each hierarchical level and the MLP is only fed by the output from the previous MLP from the second level onwards.

- CLUS-HMC [378]: A global approach based on the concept of Predictive Clustering Trees (PCT) to generate a decision tree to cover the entire tree hierarchy.

- CLUS-HMC-Ens‘[338]: This algorithm considerably improves upon CLUS-HMC by integrating a bagging strategy for creating ensembles of Clus-HMC trees.

5.4.3 Evaluation Metrics

Our framework and the competitive methods generate a probability distribution as output. Thresholding is a common practice to acquire binary prediction, but the selection of the threshold value is difficult to obtain and arbitrary. Following Wehrmann et al. [390] and Giunchiglia et al. [131], we provide quantitative evaluation using the area under the average precision-recall curve ($AU(\overline{PRC})$), whose points ($\overline{Prec}$, $\overline{Rec}$) is calculated as following:

$$\overline{Prec} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \tag{5.6}$$

$$\overline{Rec} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \tag{5.7}$$

where TP_i, FP_i, and FN_i are the number of true positives, false positives, and false negatives for class i, respectively.

In addition to $AU(\overline{PRC})$, we propose algorithms to detect global hierarchy violations. Hierarchy violations occur when the predicted probability of a child node is higher than a parent node. This definition is threshold-independent. In real applications, thresholds are required to generate predictions, and hierarchy violations occur when predictions do not include ancestors of a predicted node.

Wehrmann et al. [390] and Giunchiglia et al. [131] emphasized that a hierarchical multi-label classification model should not have any hierarchy violations and designed their models

accordingly. Wehrmann et al. [390] penalize hierarchical violation by employing a regularizer to ensure the prediction score of a node is lower than its parent nodes. Giunchiglia et al. [131] proposed a modified binary cross-entropy loss (MCLoss), which constrains the predicted probability of a child node to only be as high as its parent node. Both papers demonstrate the improvement in $AU(\overline{PRC})$ from the hierarchical structural constraints, but do not evaluate hierarchy violations. We introduce **Global Hierarchy Violation** to measure hierarchy violations.

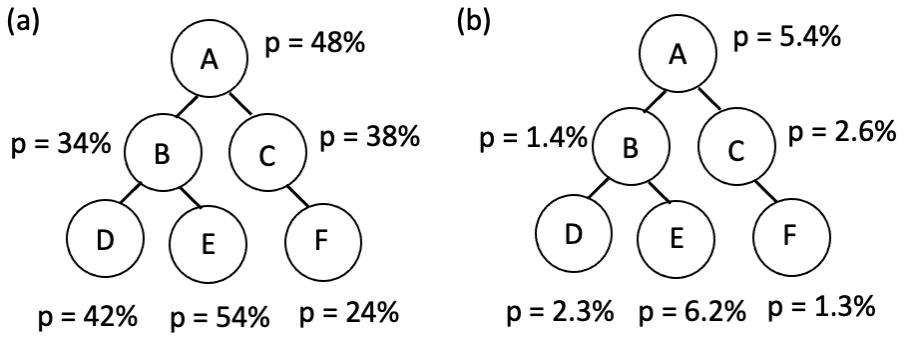


Figure 5.3: Demonstration of a hierarchy violation with predicted probabilities. Letters identify nodes and p annotations indicate predicted probabilities (pp). Hierarchy violations occur when the pp of a descendant node is higher than that of one of its ancestors. (a) B-D, B-E, and A-E pairs are hierarchy violations. (b) While the same hierarchy violation pairs are present, they are irrelevant due to low predicted confidence.

Consider the example in Fig. 5.3(a). We have a subtree of 6 nodes. The letters identify the nodes and the p annotations indicate the predicted probability scores. The right branch of this subtree has no hierarchy violation because all child nodes have lower predicted outputs than their parent nodes. On the other hand, we can observe hierarchy violations on the left branch where node B, as a parent node, has a lower score than node D and node E. Node E also has higher predicted probability than node A.

Global Hierarchy Violation compares all valid node pairs and computes the number of hierarchical violation occurrences. Given a label hierarchy $H = (V, E)$ as a tree or ontology

with labels V and edges E. Each label v_i is associated with a predicted probability p_i. A valid node pair (v_i, v_j) is defined as the shortest path between v_i and v_j does not go through the root node. Global Hierarchy Violation is defined as the total number of occurrences of the ancestor node associated with lower probability in all valid node pairs. Ancestor node is the node that has shorter shortest path to the root node in a valid node pair. For the example in Fig. 5.3(a), the algorithm would consider 8 node pairs in this subtree: A-B, A-D, A-E, A-C, A-F, B-D, B-E, and C-F, and compute the number of violation occurs in these pairs. The Global Hierarchy Violation would be 3 for this scenario. However, Global Hierarchy Violation might have a blind spot. Considering the scenario in Fig. 5.3(b), it remains a similar predicted probability pattern as Fig. 5.3(a) but with much lower probability confidence. In real life applications, most set the thresholds around 50%. The violations that happen in this scenario would be irrelevant. Global Hierarchy Violation is designed to detect all hierarchy violations and we encourage future HMC research to incorporate this metric in the evaluation purpose.

5.4.4 Implementation

To conduct a fair comparison, we adopt the code[1] provided by Giunchiiglia et al [131] for dataset pre-processing and evaluation. For pre-processing, all nominal features were converted to numeric values via one-hot encoding. The feature vectors were then normalized. All missing values were replaced by the corresponding mean. For evaluation, we remove root nodes ("root" for FUN datasets, "root", "GO0003674", "GO0005575", and "GO0008150" for GO datasets[2]). All the experiments were trained with 32 CPU cores and all reported results are the average of 10 trials. Code for all experiments will be published on github.

The hyper-parameters in our framework are the dimensions of the fully connected layers and the learning rate. These hyper-parameters are optimized with the validation set. We find that the dimension of the fully connected layers marginally impact the overall performance and the framework produces the best overall results with learning rate = 0.001. The hyper-

[1] https://github.com/EGiunchiglia/C-HMCNN

[2] https://dtai.cs.kuleuven.be/clus/hmcdatasets/

Table 5.2: Quantitative comparison with the state-of-the-art in the hierarchical multi-label classification. The numbers reported are $AU(\overline{PRC})$. Average ranking is the average of the rankings compared to other competitive algorithms among all datasets. Higher numbers are better. Our models produce superior results in 17 of 20 real-life benchmark datasets and have 1.3 average ranking.

Dataset		Ours	C-HMCNN	HMCN-F	HMCN-R	HMC-LMLP	CLUS-HMC	CLUS-ENS
FUNCAT	CELLCYCLE	**0.269**	0.255	0.252	0.247	0.207	0.172	0.227
	DERSI	**0.231**	0.195	0.193	0.189	0.182	0.175	0.188
	EISEN	**0.392**	0.306	0.298	0.298	0.245	0.204	0.271
	EXPR	**0.382**	0.302	0.301	0.300	0.242	0.210	0.271
	GASCH1	**0.369**	0.286	0.284	0.283	0.235	0.205	0.267
	GASCH2	**0.273**	0.258	0.254	0.249	0.211	0.195	0.227
	SEQ	**0.341**	0.292	0.291	0.290	0.236	0.211	0.284
	SPO	**0.241**	0.215	0.211	0.210	0.186	0.186	0.210
GO	CELLCYCLE	**0.460**	0.413	0.400	0.395	0.361	0.357	0.387
	DERSI	**0.445**	0.370	0.369	0.368	0.343	0.355	0.363
	EISEN	**0.487**	0.455	0.440	0.435	0.406	0.380	0.433
	EXPR	**0.477**	0.447	0.452	0.450	0.373	0.368	0.418
	GASCH1	**0.481**	0.436	0.428	0.416	0.380	0.371	0.415
	GASCH2	**0.473**	0.414	0.465	0.463	0.371	0.369	0.395
	SEQ	**0.478**	0.446	0.447	0.443	0.370	0.386	0.435
	SPO	**0.439**	0.382	0.376	0.375	0.342	0.345	0.372
ENRON		0.743	**0.756**	0.724	0.710	-	0.638	0.681
DIATOMS		**0.772**	0.758	0.530	0.514	-	0.167	0.379
IMCLEF07A		0.943	**0.956**	0.950	0.904	-	0.574	0.777
IMCLEF07D		0.917	**0.927**	0.920	0.897	-	0.749	0.863
AVERAGE RANKING		1.3	2.05	2.75	3.85	6.19	6.6	4.95

parameters are consistent across datasets.

5.5 Experimental Results

We report the evaluation results against the state-of-the-art models in this section. We first assess the overall performance of our model with $AU(\overline{PRC})$ against six other models in 20

real-life benchmark datasets (Sec.5.5.1). We also compute the Global Hierarchy Violation to verify that the generated predictions follow the hierarchical constraints (Sec. 5.5.2). Computation cost analysis is provided in Sec. 5.5.3. Finally, the ablation study is conducted to demonstrate the impact of the ontology learning process (Sec. 5.5.5).

5.5.1 Overall Performance vs. State-of-the-Art

Table 5.2 shows the empirical results for the current state-of-the-art models on 20 real-world benchmark HMC datasets. The results for C-HMCNN and HMC-LMLP are adopted from those published by Giunchiglia et al. [131] and results for HMCN and CLUS models are adopted from those of Wehrmann et al. [390]. For C-HMCNN, we ran their published code and verified that the results are consistent with the published results in the paper.

Our method outperforms other algorithms on 17 of 20 datasets by a significant margin. We also have the best average ranking (1.3). To demonstrate the statistical significance of the reported results, we follow previous work [390, 131] to perform Friedman test [123] and Wilcoxon Test [403]. Friedman test is a non-parametric test to compare three or more matched groups by ranks. It is used to determine if a particular factor has an effect. Wilcoxon Test calculates the difference between sets of pairs and analyze whether these differences establish statistically significant differences between the two groups. Friedman test indicates statistically significant difference with p-value of $1.06x10^{-17}$. We then apply the Wilcoxon Test to our results and C-HMCNN and it concludes that there is a statistical significant difference between the performance of our model and C-HMCNN with p-value of $3.05x10^{-05}$

To better visualize the dominant performance of our model, we create a dot chart (Figure 5.4) to show the margins to the-state-of-the-art (sota) performance. Our model outperforms all other models in FUN and GO datasets. Among the three datasets (Enron_corr, ImCLEF07A, and ImCLEF07D) that our models do not produce the best results, we still remain competitive (within 1.5%) with the best performance.

5.5.2 Hierarchy Violation Analysis

We then apply Global Hierarchy Violation to our model to examine if the predictions follow the hierarchical structure. There are no occurrences of hierarchy violations in any of the 20 datasets. We also perform Global Hierarchy Violation for C-HCMNN in FUN datasets and C-HMCNN also achieves zero hierarchy violations. This analysis shows that our model achieves the superior performance with perfect hierarchy constraints.

Figure 5.4: Margins to the state-of-the-art (sota) performance among 20 benchmark datasets. The x axis is the margin between a model performance and the sota number. We can observe that our model (red dots) demonstrates dominance among FUN and GO datasets. Our model remains competitive (within 1.5%) even when we are not the best.

5.5.3 Comparing Training Time

In this subsection, we consider training time of Surj relative to C-HMCNN[131]. C-HMCNN is assumed to be significantly faster than other competitive neural models: Unlike HMCN-

Table 5.3: Training Time Analysis. We measure the training time in seconds of our model and C-HMCNN on FUN datasets. We ran both models on a virtual machine with 32 cores. Our model is 5X to 40X faster to train.

	Ours			C-HMCNN
	Ontology Learning	classification Learning	total	total
CELLCYLE	0.8	45.3	46.1	1937
DERSI	0.4	44.2	44.4	1147
EISEN	0.3	24.4	24.7	1254
EXPR	0.4	55.7	56.1	694
GASCH1	0.3	56.8	57.1	748
GASCH2	0.6	58.6	59.2	2113
SEQ	0.2	61.3	61.5	320
SPO	0.3	44.3	44.6	1927

F, HMCN-R, and HMC-LMLP, the training time of C-HMCNN does not depend on the size of the ontology – the main contribution is a post-processing step to avoid hierarchy violations. The training time in seconds for both models on FUN datasets is shown in Table 5.3. We report results on the FUN datasets because they are smaller; C-HMCNN training is exhorbitantly expensive on larger datasets. We include training time for both ontology learning and classification learning steps. The weak correlation between the training times of the two models is due to high variance in the number of epochs needed for C-HMCNN. Our model takes significantly less time to train compared to C-HMCNN, ranging from 5x to 40x less time. In practice, ontologies change very frequently (for example, the gene ontology database releases monthly updates [3]). Low training cost allows users to adopt and integrate new ontologies more efficiently.

[3] http://geneontology.org/docs/downloads/

5.5.4 Tolerance for Evolving Ontologies

We simulate an evolving ontology by removing 20% of the leaf labels from the ontology, and removing all references to the removed leaf nodes in the training data. We consider this reduced ontology the Version 1 (V1) ontology and the corresponding training data the V1 training data. We then train our model (and a competitive baseline) as usual on V1 ontology and V1 data. Then, we restore the missing nodes to the ontology to simulate Version 2 (V2), but we do not replace the labels in the training data. That is, we now have a V2 ontology (with the 20% of nodes restored) and an (out-of-date) V1 training dataset. We relearn the ontology embeddings on the V2 ontology, and retrain the model on V1 data to simulate the situation where a new ontology version has been released, but new training data has not yet been created.

The results of this experiment are show in Table 5.4. We trained a 2-layer fully connected network with binary cross entropy loss as our naive baseline. Surj is more tolerant of the evolving graph than the baseline. As part of our future work, we are considering more realistic simulations of ontology evolution, as well as real change histories, to develop methods of improving performance on unseen data.

5.5.5 Ablation Analysis

Finally, we analyze the impact of the ontology learning process. In Table 5.5, we show results using the baseline (3 fully connected layers trained with binary cross-entropy loss and conventional one-hot encoded vectors) against our model (same baseline trained with embeddings learned from the graph-autoencoder). The performance is measured as $AU(\overline{PRC})$. Ontology learning produces significant improvement, accounting for the majority of the difference between Surj and its closest competitors.

5.5.6 Varying Data Size

Ontologies often exist in high-value applications where training data is difficult or expensive to acquire. It is important for HMC models to capture meaningful signal with minimum data provided. While the data sizes for the benchmarks are already small, we challenge Surj with

Table 5.4: Response to an Evolving Ontology. Surj is more tolerant to ontology changes than a naive baseline (in parentheses). "% changes" indicates delta of the performance on evolving ontology.

	V1 data V1 ont.	V1 data V2 ont.	% changes
CELLCYCLE	0.269 (0.211)	0.213 (0.137)	-20.8% (-35.1%)
DERISI	0.231(0.209)	0.144(0.114)	-37.6% (-45.5%)
EISEN	0.392(0.281)	0.277(0.179)	-29.3% (-36.2%)
EXPR	0.382(0.223)	0.288(0.149)	-24.6% (-33.1%)
GASCH1	0.369(0.256)	0.271(0.175)	-26.6% (-31.6%)
GASCH2	0.273(0.208)	0.213(0.118)	-21.9% (-43.2%)
SEQ	0.341(0.218)	0.242(0.146)	-29.0% (-33.0%)
SPO	0.241(0.209)	0.207(0.129)	-14.1% (-38.2%)

extreme cases to evaluate its robustness to little data environment. Figure 5.5 shows the results of varying data size environment. We train Surj with varying training data sizes, 80%, 60%, 40%, and 20%, from GO datasets. The red lines are Surj 's performance and the gray dashed lines indicate the next best model with full training data. We can see that the performance deteriorates gracefully as the data size shrinks and it remains superior in all datasets to the next best model even with only half of the training data (around 800 data records). All predictions in this experiment remain hierarchy violation-free. The robustness of Surj to low data regimes affords broad use.

5.6 Discussion

Our results show Surj outperforms competitive methods on a wide variety of datasets and is robust to low data / large ontology regimes. The design of Surj is based on a simple fully connected network and can be easily adapted to a more advanced neural network architecture targeting specific applications. For example, Surj can be attached to ResNet [151] to classify

Table 5.5: Ablation Analysis. We measure the benefit of learning ontology representations. The performance is measured in $AU(\overline{PRC})$ and we can observe that ontology learning produces significant improvement, accounting for most of the difference between our model and competitors.

	w/o ontology learning	with ontology learning	% improvement
CELLCYLE	0.211	0.269	6.7%
DERSI	0.209	0.231	10.5%
EISEN	0.281	0.392	39.5%
EXPR	0.223	0.382	71.3%
GASCH1	0.256	0.369	44.1%
GASCH2	0.208	0.273	31.2%
SEQ	0.218	0.341	21.3%
SPO	0.209	0.241	15.3%

images or BERT [92] to classify web documents, depending on the given ontology[355]. Surj has also been used by a non-profit organization to classify heterogeneous social media posts (including crawls of Twitter, Reddit, and Facebook threads) and long-form survey responses into the Sustainable Development Goals Ontology (SDG) [27] [4] and the Social Progress Index (SPI)[5]. We trained Surj on 13k discourses labeled with SPI and SDG, and achieved a F1 score of 0.353 on 2k holdout posts, compared to 0.171 with baseline multi-label classification without ontology learning. Surj is deployed within an online dashboard serving policymakers and entrepreneurs and has processed over 2M posts in specific areas. Being easy to implement and quick to train allows people with limited technical experience or access to computational resources to achieve state of the art Surj . We expect Surj to have a broader impact across different disciplines.

[4] https://www.unep.org/explore-topics/sustainable-development-goals/what-we-do/monitoring-progress/sdg-interface-ontology

[5] https://www.socialprogress.org/2020-Social-Progress-Index-Methodology.pdf

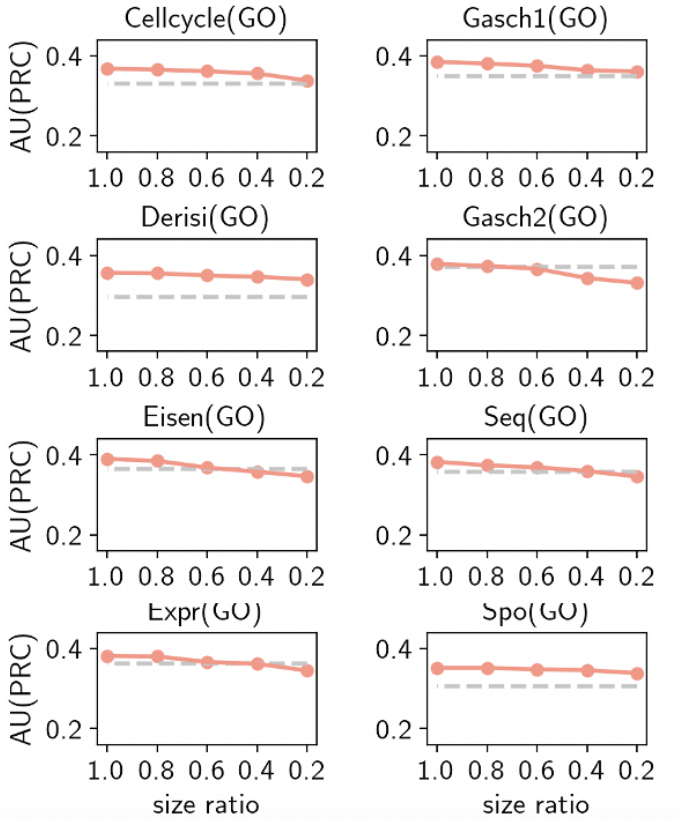


Figure 5.5: Evaluation on Surj 's robustness to varying data size. We test Surj 's sensitivity to low data scenarios. We perform the experiments using GO datasets with 80%, 60%, 40%, 20% of the training data. The gray dashed lines on the background indicate the performance of the next best model with full training data. Surj remains superior even with only half of the training data provided compared to competitors with full training data.

As future work, we intend to evaluate Surj for different data modalities, such as raw image and text. We would like to explore how Surj compares to models tailored for natural language processing [65, 67, 257, 172]. We also plan to more thoroughly investigate how Surj performs in zero-shot or few-shot scenarios. Since the relationships between labels are encoded independently of training data, Surj can predict “nearby" labels in the embedding space even without seeing labeled examples.

5.7 Ontologue: Declarative Benchmark Construction for HMC

The development of HMC algorithms [390, 131, 416] has been enabled by the availability of about 20 common benchmark datasets. These datasetes consist of protein function prediction [70, 378], annotation of medical images [95, 96], or text classification [201, 243]. While these datasets have helped inform new methods, we show that as a set they are generally small, self-similar, and can produce misleading performance results: they encode a bi-modality of an artificially challenging few-shot learning problem and an artificially easy multi-label classification problem, such that even a naive binary cross entropy solution can outperform all but the state of the art methods. In particular, we find that over 40% of labels have less than 5 data points in 16 of the 20 benchmark datasets, which constitutes a few-shot learning challenge beyond even those datasets designed to study few-shot learning [169]! Besides encouraging misleading conclusions, these datasets are artificially impoverished such that we cannot even apply state of the art few-shot learnign solutions as a baseline: the rich text associated with data items have been pre-processed into a fixed set of simple numeric features, preventing the use of modern representation learning techniques. The text associated with the label space has also been stripped away, preventing meaningful comparison with methods that can take advantage of it as a source of supervision, such as Surj [416]. Moreover, not all HMC applications constitute few-shot learning problems, so we are arguably studying the wrong problem.

We present Ontologue , a toolkit for constructing benchmarks for graph- and hierarchy-based multi-label classification problems, along with four new benchmarks derived using the system, as an evaluation tool for HMC researchers. Ontologue provides a simple declarative query interface for DBPedia [24], allowing indirect influence over the application context (any topic in Wikipedia can be used as the root), the distribution of labels and data (each label included must be associated with a minimum number of data items), and the structure of the graph (the number of hops can be specified to control the height of the hierarchy.) The resulting benchmarks represent a hierarchical (or ontological) multi-label classification task involving the assignment of labels to relevant Wikipedia abstracts. The resulting benchmarks are large, interpretable, customizable, and most importantly, exhibit statistical and

structural properties more suitable for performance analysis of state of the art methods: more balanced distribution of labels and data, fewer degenerate labels that are either trivial or impossible, availability of rich text features on both data and labels to afford multi-modal techniques, a greater proportion of DAG structures (labels with multiple parents), and a more balanced distribution of labels over the height of the tree.

5.7.1 Background

Existing Benchmarks While the HMC problem manifests in a wide variety of settings, the benchmark datasets are primarily drawn from a few specific application domains, including protein function prediction [70, 378], annotation of medical images [95, 96], and email text classification [201]. *Enron* was introduced in 2004 by Klimt et al. [201]. The inputs are feature vectors, including individual sections (From, To, Subject, and Body), and the outputs are user-specific folders. *FUNCAT* and *GO* collections are a collaborative work by Vens et al. [378] and Clare et al. [70]. The datasets within these two collections describe different aspects of the genes in yeast genome. The attributes include five types of bioinformatic data: sequence statistics, phenotype, secondary structure, homology, and expression. Both collections have identical input attributes, but the classes are taken from different knowledge graphs. FunCat is organized by the tree-structured MIPS [277] to classify the functions of gene products. The Gene Ontology (*GO*)[6] [23] is provides a DAG-structured class hierarchy, allowing terms to have multiple parents. *ImCLEF07A*, *ImCLEF07D* [95], and *Diatoms* [96] are image classification datasets introduced by Dimitrovski et al. *ImCLEF07A* and *ImCLEF07D* include Local Binary Pattern (LBP) and Edge Histogram Descriptors (EHD) for texture features and Scale-Invariant Feature Transform (SIFT) for local features of X-Ray images, but do not provide the original images. The annotations of these two datasets are part of the IRMA hierarchical classification scheme [241]. *Diatoms* is a dataset to classify images of diatoms into hierarchical taxonomic rank. The feature attributes are product of Fourier descriptors and SIFT descriptors.

[6] http://www.geneontology.org

5.7.2 Ontologue Toolkit

Ontologue is a toolkit for ontological multi-label classification dataset construction from DBPedia. This toolkit allows users to control contextual, distributional, and structured properties and create customized datasets. We also further provide four benchmark datasets that better represent the problem space from the toolkit and implement the existing HMC algorithms to provide baselines.

5.7.2.1 Benchmark Requirements

We designed Ontologue to offer declarative influence over three properties that can influence performance of various methods. We intentionally designed Ontologue to limit explicit, low-level control over benchmark properties to discourage the creation of cherry-picked datasets that would invalidate generalizability claims.

Application Context Existing benchmark datasets are of a fixed and sometimes opaque domain, complicating interpretation and preventing the use of multi-modal methods. For example, the Gene Ontology datasets have been stripped of information: node labels have been replaced with integers, and data item features may be pre-processed using outdated feature engineering techniques. While this approach is reasonable to isolate and demonstrate the ability of proposed methods to exploit the hierarchy, we lose the ability to use modern representation learning methods that offer a less naive baseline. Moreover, qualitative assessment of the performance is impossible. Ontologue can produce a customized benchmark based on any topic in Wikipedia, and includes node labels and rich data features (Wikipedia abstracts and metadata) to admit appropriate baselines and encourage models that can exploit all available sources of supervision.

Distribution of labels and data With large hierarchies (up to 4000 nodes), the distribution of the labels over the data becomes significantly imbalanced, reducing the problem to an artificially challenging few-shot learning problem in the long tail and an artificially easy multi-label classification problem in the head. Ontologue offers control over the distribution by removing any label that is associated with fewer than a given threshold of data items.

Hierarchy Structure A hierarchy of labels typically models specificity in the underlying universe of discourse: general categories are at the top of the hierarchy and specific categories are at the bottom. As a result, the height and width of the tree can influence the distribution, and therefore performance. Ontologue affords control over the number of hops from the root to offer declarative control over the size, gross graph structure (i.e., height/diameter), and distribution (since distribution correlates with height via specificity).

Also, ontologies are generally DAGs rather than strict hierarchies, and methods that require a hierarchy must pre-process the graph to duplicate children with multiple parents. Although we did not find that this pre-processing step significantly influenced performance, a benchmark should include a significant number of nodes with multiple parents to reflect realistic ontologies. Ontologue assumes the source data and the resulting benchmarks are DAGs to avoid artificially restricting the space.

5.7.2.2 Declarative Benchmark Generation

We designed Ontologue as a simple query interface over DBPedia. [24]. DBPedia[7] [24] is a crowd-sourced community project to extract structured content from several Wikimedia projects[8]. The result is a large multi-domain ontology consisting of 3.77 million "things" with 400 million "facts." The structured information is organized as an Open Knowledge Graph and is publicly available. Variations of the DBPedia dataset have served as benchmarks for a variety of machine learning research, such as text classification [406, 417], question answering [181], and information retrieval [366, 285]. DBPedia is under Creative Commons Attribution-ShareAlike 3.0 License and does not include personally identifiable information or offensive content .

We use the collection from the snapshot for December 2021 [9]. We collect the Wikipedia Article Categories adhering to the Simple Knowledge Organization System (SKOS) schema [177]. The entire graph includes 1,713,451 entities and 4,298,587 edges.

---

[7] https://www.dbpedia.org/

[8] https://www.wikimedia.org/

[9] https://databus.dbpedia.org/dbpedia/collections/dbpedia-snapshot-2021-12/

Algorithm Starting from a root concept `TOPIC`, we perform a depth-first search following `skos:broader` relationships between terms to build the label space, and then include all Wikipedia abstracts that use at least one of those labels. Any label that is used in fewer than `MINDATA` data items is ignored, along with its data. The search stops after `MAXHOPS`. The `skos:broader` relationship can include multiple parents for a concept, such that the result may be a directed acyclic graph (DAG).

With these three parameters `TOPIC`, `MINDATA`, and `MAXHOPS`, Ontologue allows benchmark creators declarative, high-level control over context, distribution, and structural properties of the generated datasets, respectively.

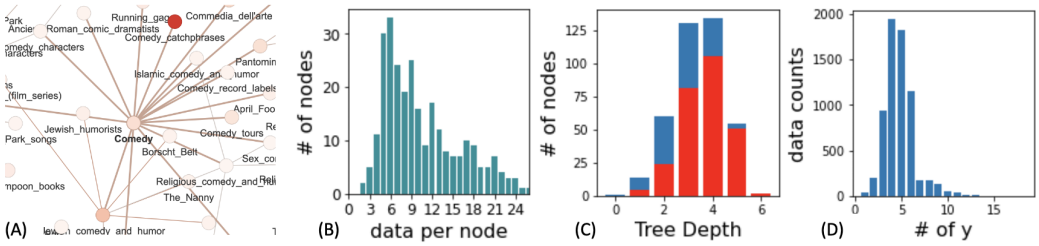


Figure 5.6: An illustration of the visual analysis features in Ontologue . (A) An interactive graph visualization affording qualitative review of the contextual and structural properties of the derived label hierarchy. (B) A histogram of data distribution over labels. (C) The distribution of easy and hard labels over tree depth. (D) The distribution of the number of labels per data item.

Visual Analysis To use Ontologue effectively, the users need to be able to visualize and analyze the results. Ontologue provides two interactive visualizations for evaluating the properties and suitability derived benchmarks: a graph viewer and a distribution viewer. Figure 5.6 is an illustration of the available visualizations offered by Ontologue . On the left is an interactive graph viewer. The users are provided with options to color code the nodes with given properties and are able to navigate the graph interactively to investigate the context and the structure of the label hierarchy. The interactive graph viewer is created

with Pyvis package[10]. The distribution viewer (on the right of Figure 5.6) gives the users various of ways to determine the distributional properties of the generated benchmarks. Ontologue does not provide users total explicit control over a benchmark to avoid cherry picking, but a declarative way to create customized benchmarks.

Example Benchmarks We derived four example benchmark datasets for the topics `Comedy`, `Engineering`, `Law`, and `Main`. `Law` is generated with "Law" as a start node with 10 hops and 30 minimum data points. `Comedy` is created with "Comedy" as the anchor and is allowed 6 hops and minimum 5 data items per class label. To develop `Engineering`, we set maximum hop as 7, minimum data as 6, and "Engineering" as the start node. Finally, `Main` is achieved with 7 hops and 70 minimum data points from the top of the DBPedia graph. The statistics can be seen in Table 5.6. We will discuss the properties of the derived benchmarks in the next section.

5.7.3 Performance Results

In this section, we summarize existing benchmarks and four proposed benchmarks derived using Ontologue . We then report on performance results, showing that the proposed benchmarks can change the ranking of competing methods.

5.7.3.1 Benchmark Statistics

Table 5.6 shows statistics of the current and proposed benchmarks. We make several observations:

Size We observe that the prior datasets are relatively small in the number of data items, ranging from 1k to 11k. The four proposed datasets, in contrast, range in size from 8k to 61k.

Domain and Structural Diversity Among the prior datasets, there is low diversity in the hierarchies: While the four small datasets cover diverse domains, they are struc-

[10] `https://pyvis.readthedocs.io/en/latest/index.html`

turally less interesting. There are only two larger, more interesting hierarchies: all *FUNCAT* datasets share the same label hierarchy and all *GO* datasets also share the same label hierarchy. While many papers report on a variety of these "sub" datasets, they are semantically and structurally similar, undermining generalization arguments. The set of four proposed datasets, in contrast, exhibits diversity in size of the data, size of the hierarchy, and the depth of the hierarchy. Moreover, the inputs of all prior datasets are pre-processed feature vectors. Modern methods using learned embeddings provide another source of supervision that can be an appropriate baseline or inform new methods; ignoring this context artificially constrains the solution space.

Label Distribution: HMC or Few-shot learning? The prior datasets are often severely class-imbalanced [131], as shown by the average and median number of data items per label. The long-tailed distribution of the number of data items per label suggest that the challenge of these datasets is not in the hierarchical structure, but in the embodiment of a conventional few-shot learning problem.

To further investigate this issue, we calculated the percentage of nodes with lower than 5 pieces of data points relative to the whole hierarchy and Table 5.6 shows the result. For reference, current benchmarks for few-shot classification are MiniImagenet [382], CIFAR-FS [34], and CUB [383]. The MiniImagenet, a sub set of the Imagenet[220], includes 100 classes and 600 images per class. The standard procedure is to split into 64 classes for training, 16 novel classes for validation and 20 novel classes for testing. The CIFAR-FS dataset also has 100 classes with 100 images per class. The set up for few shot learning is the same as the MiniImagenet. The CUB dataset has 200 classes with 11,788 images in total (averaged 58.9 images per class). The dataset is split into 100 base classes, 50 novel classes for validation, and 50 novel classes for testing [169]. The percentage of novel classes to all classes in the test set for MiniImagenet, CIFAR-FS, and CUB are 20%, 20%, and 25%, respectively.

Some of these HMC benchmark datasets are arguably more difficult than standard few-shot benchmark datasets: (i) There are a higher number of few-shot classes in *DIATOMS*, the *FUNCAT* collection, and the *GO* collection. (ii) There is less data available per class for training in all but 1 HMC datasets than few-shot datasets (iii) Few-shot learning algorithms

often rely on alternative sources of supervision or significant pre-training to predict novel classes. These resources are not accessible in HMC benchmarks due to the limited information of the datasets. While it is practical that limited data is available for large amount of labels in the real-life HMC problems, the few-shot aspect of HMC problems has never been explored in existing HMC studies. We believe a clear characterization of the HMC benchmark datasets will provide a novel direction and facilitate HMC research.

Table 5.6: Statistics of current benchmark and proposed datasets. "# Classes" indicates number of labels in each dataset. "# data" means number of data items. "Averaged #/class" implies average number of data items available per class. "Median #/class" suggests median number of data items available per class. "5-shot" means the percentage of labels with less than 5 data items. We use 'cellcycle' to represent FUNCAT and GO collections.

Taxonomy	Dataset	# Classes	Depth	# data	Averaged #/class	Median #/class	5-shot
DBPEDIA (DAG)	COMEDY	395	6	8333	80.77	13.00	20.5%
	ENGINEERING	587	7	22735	226.25	31.00	14.3%
	LAW	958	10	61974	616.39	72.50	0%
	MAIN	147	7	15310	529.18	139.00	0%
Tree	ENRON	56	3	1648	90.2	16.0	21.4%
	DIATOMS	398	3	2603	10.1	5.0	48.5%
	IMCLEF07A	96	3	11006	312.5	119.5	9.4 %
	IMCLEF07D	46	3	11006	652.2	137.0	19.6%
DAG	FUNCAT	499	6	3757	28.5	7.0	43.7 %
	GO	4122	12	3751	12.5	2.0	71.7%

5.7.3.2 Evaluation of State of the Art Methods

We report performance of a naive baseline and several state of the art methods against the four proposed benchmarks.

Naive Baseline is a three-layered feed forward network optimized with binary cross entropy. This network is similar to basic multi-label classification model. **Surj** is the first HMC algorithm integrating Graph Neural Network in its framework. Surj learns a representation of the label hierarchy with a graph autoencoder and maps the input onto the representation space with cosine similarity. The network is also optimized with binary cross entropy. **C-HCMNN**[11] [131] is designed to produce coherent predictions which respect hierarchical structure. The network adopts a modified binary cross entropy loss that enforcing the predicted probabilities of children nodes are lower than their ancestors. **HMC-GA**[12] [61] is a global method too induce HMC rules which are generated using a deterministic procedure considering the classes assigned to instances. **CLUS-HMC**[13] [378] is based on Predictive Clustering Trees (PCT) to generate a decision tree to cover the entire tree hierarchy. **CLUS-HMC-ENS** [338] significantly improves upon CLUS-HMC by using a bagging strategy to create ensembles of Clus-HMC trees.

AUPRC We use standard metric for HMC $AU(\overline{PRC})$ to evaluate the performance of baseline methods on the four benchmarks. $AU(\overline{PRC})$ is universally adopted [390, 416, 131] to evaluate the performance of HMC algorithms. $AU(\overline{PRC})$ is defined as the area under the average precision and recall curve. Utilizing $AU(\overline{PRC})$ allows studies to avoid thresholding, in which the selection of the threshold can be application-dependent, arbitrary and difficult to acquire.

Global Hierarchy Violations We also measure the compliance with hierarchy constraints using the Global Hierarchy Violations (GHV) measure introduced in [416]. HMC methods assign a probability to each label. A violation occurs when an ancestor is assigned

[11] https://github.com/EGiunchiglia/C-HMCNN

[12] http://www.biomal.ufscar.br/hmc.html

[13] https://dtai.cs.kuleuven.be/clus/

a lower probability than its descendant. GHV is the count of all such violations.

Many HMC models are designed to avoid hierarchy violations. Wehrmann et al. [390] use a regularizer to penalize hierarchical violations and Giunchiglia et al. [131] employ a modified binary cross-entropy loss (MCLoss), which constrains the predicted probability of a descendant node to only be as high as its ancestors. While both algorithms improved upon the state of the art in $AU(\overline{PRC})$, both studies failed to demonstrate whether any hierarchy violations actually occurred. Global Hierarchical Violations was proposed as a measure to address this issue [416].

Implementation For each dataset, we run all algorithms 10 times and calculate the averaged $AU(\overline{PRC})$ and the normalized Global Hierarchical Violation. The normalized Global Hierarchical Violation computes the Global Hierarchical Violations over all valid node pairs. All algorithms do not consider auxiliary information (such as text label for each node) of the label hierarchy for fairness. Ontologue and the four proposed datasets will be available on GitHub[14].

Table 5.7 shows the results of all methods on the four new proposed benchmarks. We see that unlike prior results, which showed that Surj was superior on all but three datasets, there is more diversity in the ranking. We conclude that these benchmarks can reveal limitations to generalization and inform research in new classes of methods. We note, however, that when Surj is free to use the node labels as a source of additional supervision (Table 5.8), it remains the state of the art.

5.7.4 Comparative Analysis of Benchmarks

In this section we perform additional experiments to study the quantitative impact of improving benchmark diversity. We consider requirements around context, label distribution, and structure.

Application Context Because our benchmarks provide context, e.g. text associated with the labels, methods can take advantage of this additional information. Surj framework is

[14] https://github.com/seanyang38/Ontologue

Table 5.7: $AU\overline{PRC}$ and normalized Global Hierarchy Violations (parentheses) on proposed benchmarks for several state of the art methods and a naive baseline based on binary cross entropy loss. The naive baseline outperforms all but the top two methods, suggesting that some previous results were confounded by label distribution issues.

	COMEDY	ENGINEERING	LAW	MAIN
Naive	0.839 (0.013)	0.829 (0.007)	0.777 (0.019)	0.782 (0.038)
Surj	0.857 (0.005)	**0.853 (0.003)**	**0.796 (0.008)**	0.809 (0.006)
C-HMCNN	**0.870 (0.004)**	0.842 (0.005)	0.783 (0.010)	**0.819 (0.006)**
HMC-LMLP	0.627 (0.012)	0.730 (0.016)	0.671 (0.019)	0.663 (0.018)
CLUS-HMC	0.620 (0.023)	0.603 (0.035)	0.632 (0.025)	0.640 (0.020)
CLUS-ENS	0.713 (0.014)	0.695 (0.027)	0.693 (0.013)	0.734 (0.013)

Table 5.8: Surj performance ($AU\overline{PRC}$ (GHV)) with and without text labels. Performance improves on all datasets when text labels are available, making Surj the state of the art since other methods cannot make use of the text.

	COMEDY	ENGINEERING	LAW	MAIN
Surj w/ label text	0.872 (0.004)	0.873 (0.002)	0.802 (0.004)	0.834 (0.005)
Surj w/o label text	0.857 (0.005)	0.853 (0.003)	0.796 (0.008)	0.809 (0.006)

able to consider label features. We analyze whether this contextual information improves the performance of Surj on the proposed benchmarks. We embed label text with BERT and serve the embeddings as node features during graph auto-encoder training. We run the rest of the framework with the same procedure. Table 5.8 shows the performance on Surj with and without considering label text. Surj improves on all datasets for both $AU(\overline{PRC})$ and normalized Global Hierarchical Violation. It implies that contextual information from the graph can help HMC problem.

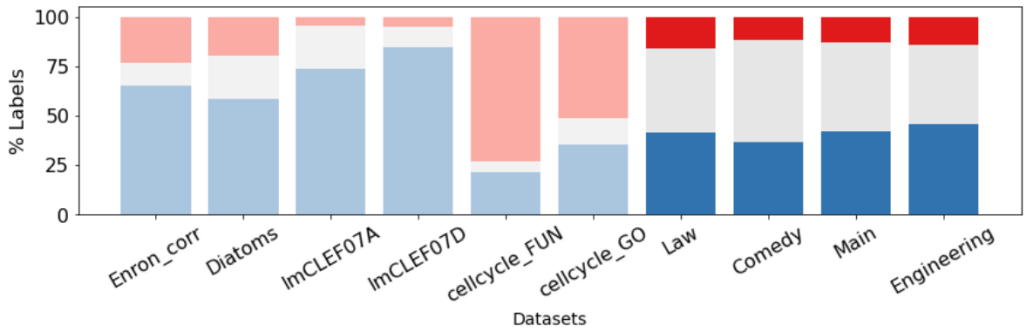


Figure 5.7: The percentage of "trivial" labels for which all methods are successful (blue), "impossible" labels for which no method is successful (red), and all other labels (gray). Ontologue benchmarks have a significantly lower proportion of trivial and impossible labels (45% to 58%), and are therefore more useful for analyzing performance and generalizability.

Distribution Meding et al. [276] find that ImageNet validation set suffers from dichotomous data difficulty (DDD): 57.5 % of the images are either "trivial" (too easy) or "impossible" for the model to recognize. We adopt similar methodology used in the study to analyze the difficulty of the current and proposed benchmarks. Instead of looking at data items, we consider the difficulty of the labels. We compute $AU(\overline{PRC})$ for each label for all 6 baseline algorithms. The label is regarded as "trivial" when 5 or more models have over 0.5 $AU(\overline{PRC})$. On the other hand, the label is considered as "impossible" when 5 or more models have less than 0.5 $AU(\overline{PRC})$. We calculate the percentage of "trivial" and "impossible" labels from all datasets and plot the results in Figure 5.7. Red/Pink indicates "impossible" examples, and blue means "trivial" labels, and gray shows interesting cases where the current methods have a variety of performance. The current benchmarks have over 75 % of labels are either too easy or impossible for the models to solve, while the proposed benchmarks range from 45 % to 58 %. This analysis implies that the proposed benchmarks provide diversity and create confusion between models.

Structure Finally, we analyze the structural property of the proposed benchmarks. The left chart of Figure 5.8 shows the average $AU(\overline{PRC})$ by tree depth and the right chat

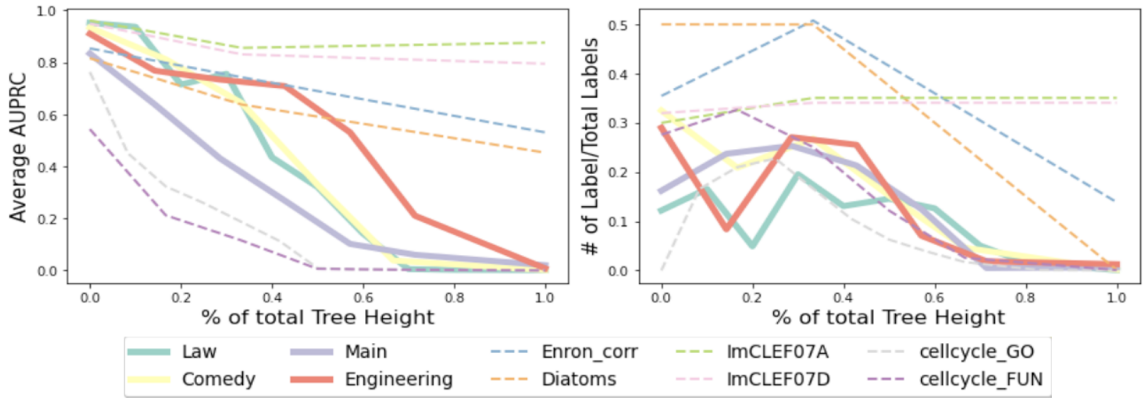


Figure 5.8: Left: Average $AU(\overline{PRC})$ over tree depth. Performance is influenced by tree height. Right: Distribution of labels over tree depth. Performance of GO and FUNCAT benchmarks are dominated by low-coverage labels at the bottom, while other benchmarks are shallow and therefore unrealistically independent of tree structure. Ontologue datasets (in bold) offer a smooth decline over hierarchy depth (left) and a more balanced distribution (right).

demonstrate the distribution of labels over tree depth. The experimental results are average of naive baseline, Surj, and C-HMCNN. We can observe in the left chart that the current benchmarks are extreme: the depth of the trees has either limited impact (four dashed lines on the top) or significant impact (two dashed lines at the bottom) to the performance. The proposed benchmarks (solid thicker lines) demonstrate a more smooth decline over tree depth and this presents a more reasonable problem space for HMC.

5.7.5 Discussion

In this study, we demonstrate that current HMC benchmarks exhibit degenerate statistical and structural properties that can confound performance analysis among competing methods. We present a declarative toolkit, Ontologue , for generating custom, realistic, and diverse benchmarks that can better assess claims of generalizable performance. We use Ontologue to derive four new benchmarks using DBPedia and show that these benchmarks

provide application context that can afford multi-modal approaches, more representative distribution of data over labels, and more realistic structural properties. We also recognize some limitations of our work: Our datasets are based on a single (though diverse) source of text, and the declarative interface could afford cherry-picking designs through trial and error. We conclude that these benchmarks and the Ontologue system itself will improve the diversity of representative tasks and continue to lead to new insights in the analysis of graph-oriented labeling tasks.

5.8 Summary

While large-scale labeled data is uncommon in high expertise domains, human attention tend to invest in curating ontology to describe complex relationships. Although these relationships between terms provide some supervision, this additional information is often ignored by learning solutions. Hierarchical multi-label classification aims to address this problem. In this chapter, we propose Surj, which learns a representation of the label hierarchy and separately learns to map input records onto learned representation space of the label hierarchy and produce predictions. This framework reduces dependence on the training data and can make predictions even for underrepresented labels. We then introduce Global Hierarchy Violation to measure the hierarchy violations of the predicted outputs. We also discover that the current HMC benchmarks suffer from several deficiencies: (i) high percentage of labels have little data to train. The HMC problem is arguably harder than the few-shot problem. (ii) The datasets are outdated and lack of context. (iii) Although there are 20 common used benchmarks, 16 of them share two label hierarchies. It raises concerns for generalizability. In response, we propose Ontologue, a declarative declarative query system for generating custom benchmarks with specific properties, then use this system to design 4 new benchmarks extracted from DBPedia that better represent the problem space. We hope our contribution in a novel model, a metric to measure hierarchy violations, and new benchmarks can facilitate HMC research.

5.9 Supplementary Materials

We report additional data and experimental results. We elaborate Global Hierarchical Violation in Section **??**, offer a snippet of the DBPedia graph in Section 5.9.0.1, and provide supplementary comparative analysis of benchmarks in Section 5.9.0.2. Checklist is provided in the last page.

5.9.0.1 A Snippet of the DBPedia Graph

To offer a general impression of the DBPedia graph, we provide a snippet of the top levels of the DBPedia graph. Unlike most knowledge bases which only cover specific domains, DBPedia, as shown in Figure 5.9, includes a wide variety of domains and represent real community agreement. The ontology carries 3.77 million "things" with 400 million "facts."

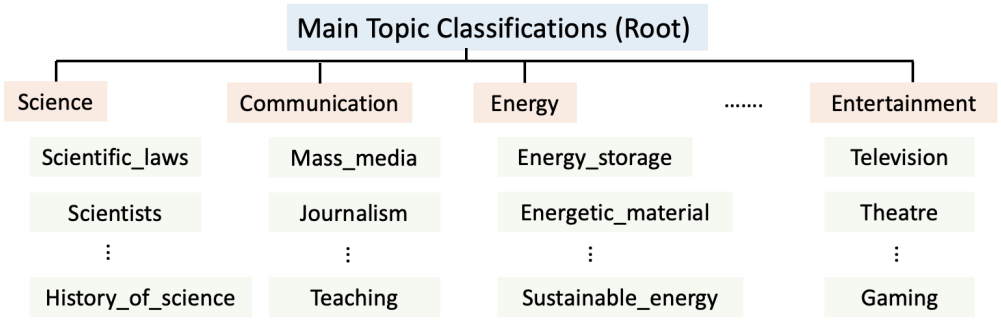


Figure 5.9: A snippet of the top levels of the DBPedia graph.

5.9.0.2 Additional Comparative Analysis of Benchmarks

We demonstrate supplementary comparative analysis of the current and Ontologue benchmarks.

Statistics of the subsets in GO and FUNCAT collections Table 1 in Section 4 shows combined statistics of proposed and current benchmarks. We select *Cellcycle* to represent *FUNCAT* and *GO* collections because they share similar data statistics and same graphs.

Table 5.9: Data statistics of **FUNCAT** and **GO** collections. The collections are lack of diversity and undermine generalization arguments.

Taxonomy	Dataset	# Classes	# Attributes	Depth	# data
FUNCAT (Tree)	CELLCYCLE	499	77	6	3757
	DERSI	499	63	6	3725
	EISEN	461	79	6	2424
	EXPR	499	551	6	3778
	GASCH1	499	173	6	3764
	GASCH2	499	52	6	3779
	SEQ	499	478	6	3919
	SPO	499	80	6	3703
Gene Ontology	CELLCYCLE	4122	77	12	3751
	DERSI	4116	63	12	3719
	EISEN	3570	79	12	2418
	EXPR	4128	551	12	3773
	GASCH1	4122	173	12	3758
	GASCH2	4128	52	12	3773
	SEQ	4130	478	12	3900
	SPO	4116	80	12	3697

Table 5.9 reveals the complete statistics of the subsets in the *FUNCAT* and *GO* collections. All *FUNCAT* datasets share the same label hierarchy and all *GO* datasets also share the same label hierarchy. While many papers report on a variety of these "sub" datasets, they are semantically and structurally similar, undermining generalization arguments.

Distributional Analysis We perform four data distributional analysis on current benchmarks (Figure 5.11) and proposed benchmarks (Figure 5.10): (i) Data availability for labels (first column) (ii) Data distribution across hierarchy levels (second column) (iii) Node and

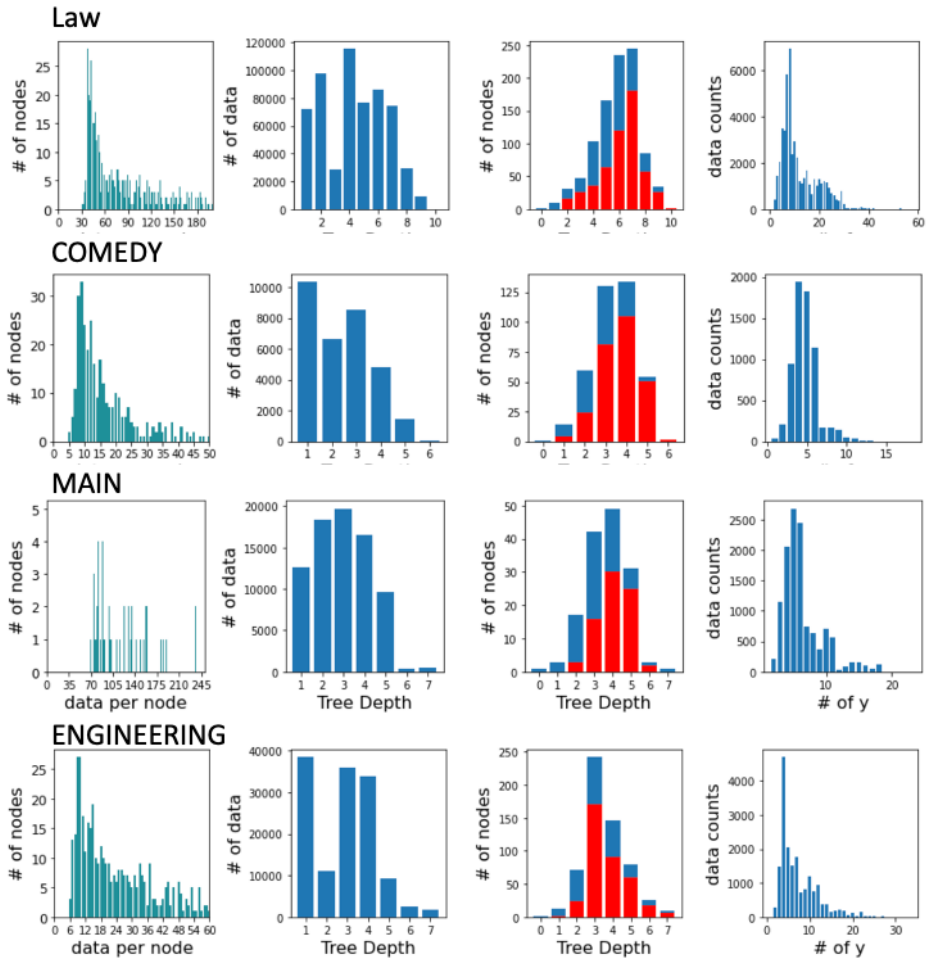


Figure 5.10: Distribution analysis for the proposed benchmarks. The first column is the data availability for labels. The second column shows the data distribution over the label hierarchy levels. The third column demonstrates the node distribution over hierarchy levels with red bars indicate leaf (nodes without children) distribution over hierarchy levels. The forth column illustrates label distribution over data.

leaf (marked in red) distribution across hierarchy levels with (third column) (iv) Label distribution over data (forth column).

- **Data availability for labels (first column)**: Each chart shows a histogram describing data availability for each node for the corresponding dataset. The histograms

confirm that HMC datasets are often class imbalanced [131]. Significant number of nodes in *Diatoms*, *FUN* collection, and *GO* collection have less than 5 data points to train on. The long-tailed distribution of the number of data items per label suggest that the challenge of these datasets is not in the hierarchical structure, but in the embodiment of a conventional few-shot learning problem. Our proposed toolkit and benchmarks address this issue by setting minimum data availability for the labels. This provide proper benchmarks for HMC algorithms to reduce the influence of the few-shot problems and evaluate the performance over hierarchical structure.

- **Data distribution across hierarchy levels (second column)**: Nodes at deeper levels often have lower data availability compared to nodes at higher levels as we can observe in *DIATOMS*, *FUNCAT*, *GO* collections and in proposed benchmarks. Our proposed benchmarks still carries high data availability at the deep levels.

- **Node and leaf distribution across hierarchy levels (third column)**: The label hierarchies of *ENRON*, *DIATOMS*, *IMCLEF* are shallow and the leaves (nodes without children) are all located in the bottom two levels. This undermines the difficulty of HMC problem provided from the label hierarchy. The proposed benchmarks have more organically constructed label hierarchies with leaves range from level 2 to level 10.

- **Label distribution over data**: We discover that three of the datasets, *DIATOMS*, *ImCLEF07A*, and *ImCLEF07D* are not necessarily multi-label datasets because every data point is only assigned to one branch of labels. This significantly reduces the difficulty of the problem.

The current benchmarks can be seen as tri-modal: There are three small and shallow hierarchies (*ENRON*, *DIATOMS*, and *IMCLEF*) from diverse domains with less than 100 nodes, a related set of functional genomic hierarchies of moderate size (*FUNCAT*), and gene ontology hierarchies where the number of labels is approximately the same as the number of data items *GO*. Since the size and structure correlates with domain, it becomes difficult to determine how performance results may generalize — we do not have large hierarchies in

social domains or small hierarchies in biological domains, for example. Ontologue datasets provide diverse data distributions and tree structure to improve the generalization of the HMC benchmarks.

Structural Analysis We investigate the structure of the label hierarchies by computing the averaged number of children and parents. Table 5.10 demonstrates the comparison between current benchmarks and Ontologue datasets. We disregard the root node when we calculate the average number of parents and ignore the leaves when we compute the average number of children. Label hierarchies in most current benchmarks are trees, so the number of parents for all nodes is 1. *Enron* is an extreme case given that there are only three internal nodes, not a root or leaf, in the label hierarchy. We can observe that the Ontologue datasets carry more complex and diverse hierarchies. The complexity of the label hierarchies provides a more difficult problem space for HMC.

Table 5.10: Average number of parents (not considering roots) and children (not considering leaves) of the current and proposed benchmarks. Label hierarchies in most current benchmarks are trees, so the number of the parents for all node is 1. Only three nodes in the *ENRON* label hierarchy are internal nodes (not a root or leaf) and they carry a lot of children nodes. The label hierarchies in the Ontologue datasets are more diverse and complex and provide a more difficult HMC problem space.

Taxonomy	Dataset	Averaged # of parents	Averaged # of children
DBPEDIA (DAG)	COMEDY	1.33	2.96
	ENGINEERING	1.14	3.52
	LAW	1.29	2.68
	MAIN	1.52	4.12
Tree	ENRON	1.00	15.00
	DIATOMS	1.00	3.12
	IMCLEF07A	1.00	2.66
	IMCLEF07D	1.00	2.11
DAG	FUNCAT	1.00	2.71
	GO	1.41	2.78

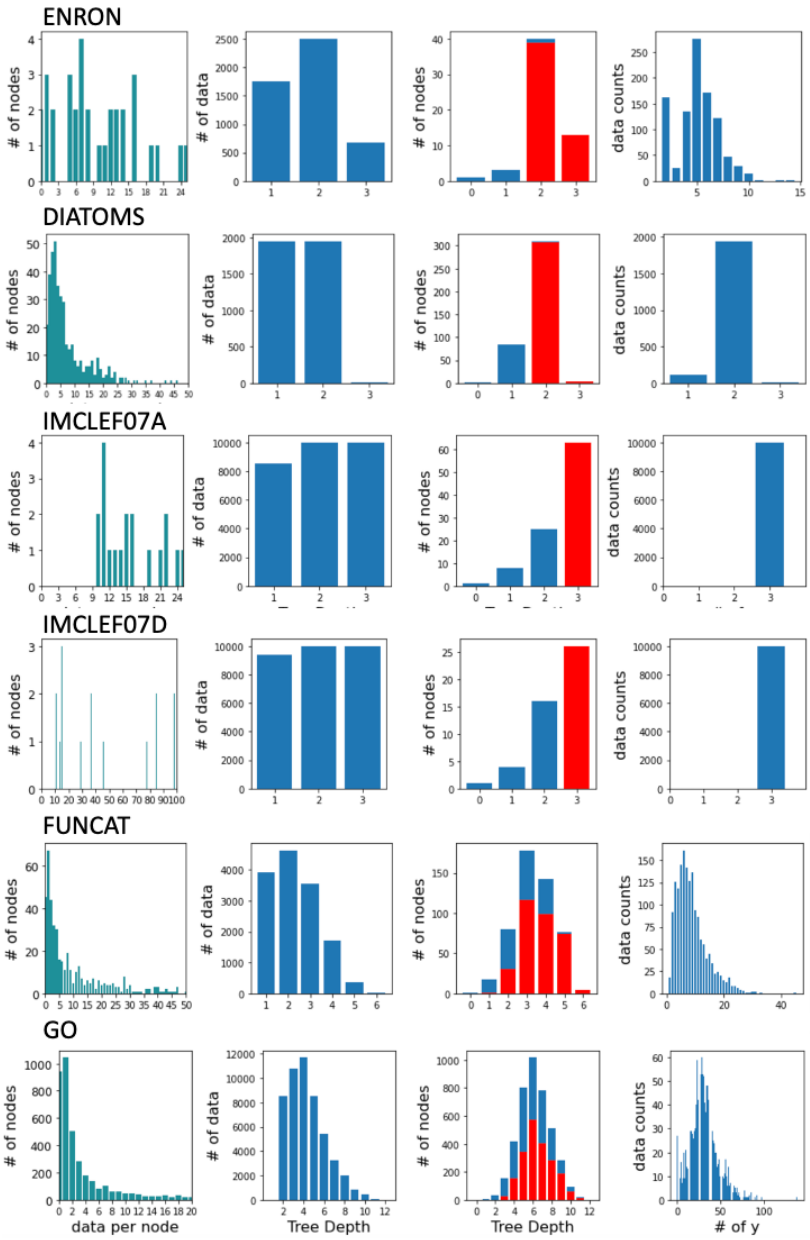


Figure 5.11: Distribution analysis for the current benchmarks. The first column is the data availability for labels. The second column shows the data distribution over the label hierarchy levels. The third column demonstrates the node distribution over hierarchy levels with red bars indicate leaf (nodes without children) distribution over hierarchy levels. The forth column illustrates label distribution over data.

Chapter 6

METHODS FOR EXPLORATORY ANALYSIS FOR LARGE-SCALE HETEROGENEOUS IMAGE-TEXT COLLECTIONS.

6.1 Introduction

We consider multi-modal unsupervised learning for image-text pairs. In many science and engineering applications, images are equipped with free-text descriptions, but structured training labels are difficult to acquire. For example, the figures in the scientific literature are an important source of information [240], but training labels require specialized knowledge and may evolve frequently. These figures are, however, equipped with a caption describing the content or purpose of the figure, and these captions can be used as a source of (noisy) supervision. Other examples include medical imagery (paired with unstructured physician's notes), art (paired with the artist's or curator's description), or archaeological artifacts (paired with researcher's notes).

A simple approach is to ignore the text and cluster the images alone. Unsupervised image clustering has received significant research attention in computer vision [405]. However, as we will show, these single-view approaches fail to differentiate semantically different but visually similar subjects on benchmark datasets. On the other hand, using the captions alone (ignoring the image) is rarely considered viable, since captions do not fully describe the content of the image. Current multi-modal image-text models focus on matching images and corresponding captions for information retrieval tasks [192, 101] rather than unsupervised learning for image-text pairs. Jin et al. [182] characterize correlations between image and text using Canonical Correlation Analysis (CCA). However, the textual information for the model comprised semi-structured tags rather than unstructured free-text descriptions. Free-text descriptions can capture more information to improve a model's performance, but unstructured text is often considered prohibitively noisy in practice: it can contain irrelevant or inconsistent information, and may even be associated with the wrong image. We find that

these challenges have limited the uptake of machine learning in complex, human-intensive domains in science and the humanities.

We propose *JECL*, a clustering algorithm for image-text pairs that considers both visual features and text features, learning both a vector representation for the pair as well as a clustering. JECL extends prior work on Deep Embedded Clustering (DEC) [405]. DEC learns a mapping function from the data space to a lower-dimensional feature space and produces soft cluster assignments in which it iteratively optimizes Kullback-Leibler (KL) divergence between soft assignments and computed target distributions. DEC has shown success on clustering several benchmark datasets including both images and text (separately). Despite its utility, we find DEC may often generate empty clusters or singleton clusters containing an obvious outlier, a common problem in clustering tasks [97].

JECL learns cluster assignments by iteratively optimizing a clustering objective while learning to align a text distribution **r** and an image distribution **q**. We address the problem of empty and singleton clusters by introducing regularization terms to force the model to find a solution with a more balanced assignment for each track. We design a target distribution **p**, such that the model learns by minimizing the KL divergence between **q** and **p**, the KL divergence between **r** and **p**, and the Jensen-Shannon Divergence between **r** and **q**, penalizing the model when these distributions become dissimilar. The final cluster assignments are derived via softmax over the joint distribution (Figure 6.1). These combined objectives help JECL define clear boundaries between clusters in the embedding space while retaining semantically meaningful results. In contrast, DEC can fail to differentiate between visually similar but semantically distinct examples, as we will show.

6.2 Related Work

We consider related work in both image-text representation learning and multi-view clustering methods.

Multi-View Image-Text Representation DeVise [124] generates visual-semantic embeddings by linearly transforming a visual embedding from a pre-trained deep neural network into the embedding space of the text representation. After DeVise, several visual semantic

models have been developed by optimizing bi-directional pairwise ranking loss [199, 386] and maximum mean discrepancy loss [370]. Maximizing CCA (Canonical Correlation Analysis) [143] is also a common way to acquire cross-modal representation. Yan et al. [411] address the problem of matching images and text in a joint latent space learned with deep canonical correlation analysis. Dorfer et al. [101] develop a canonical correlation analysis layer and then apply pairwise ranking loss to learn a common representation of image and text for information retrieval tasks. However, most image-text multi-modal studies focus on matching image and text. Few methods study the problem of unsupervised clustering of image-text pairs.

Jin et al. considered clustering images by integrating the multimodal feature generation with a Locality Linear Coding (LLC) and co-occurrence association network, multimodal feature fusion with CCA, and accelerated hierarchical k-means clustering [182]. However, the text data they handled are tags instead of the long, noisy, and unreliable free-text descriptions we are interested in. Grechkin et al. proposed EZLearn [135], a co-training framework which takes image-text data and an ontology to classify images using labels from the ontology. This model requires prior knowledge of the data in order to derive an ontology. This prior knowledge is not always available, and can significantly bias the results toward the clusters implied by the ontology.

Multi-View Clustering JECL can be considered as a form of multi-view clustering, except that multi-view methods often only consider only one data type, typically multiple images of the same object. Matrix factorization is a common approach to address the multi-view clustering problem. Liu et al. [254] used a joint matrix factorization with restraints to progressively find the consensus between different views. Zhao et al. [424] developed a deep matrix factorization framework, which imposes the non-negative representation of all views to be the same in final layer to maximize the mutual information and is graph regularized to preserve the local geometric structure in each view. BMVC, binary multi-view clustering, is presented by Zhang et al [423] to easily scale to large data by collective encoding views into a compact common binary code space and simultaneously clustering the collaborative binary representations using a matrix factorization model. Spectral clustering has shown significant

performance in single view tasks and it also has been explored in multi-view scenario. Brbić et al. [49] proposes a multi-view spectral clsutering framework that encourages sparsity and low-rank solution and balances the agreement across views. However, most spectral clustering methods are not scalable due to its quadratic complexity.

JECL is also robust to incomplete multi-view tasks, which have received increasing attention in recent years. DAIMC [168] is a method built on weighted semi-nonnegative matrix factorization. It learns a shared feature matrix for all views and prevents the effect from missing view with $L_{2,1}$-Norm regularized regression. Wang et al. [385] build a bridge between perturbation risk bounds and missing view problems and propose PIC which reduces perturbation risk among all views and learns a consensus Laplacian matrix.

6.3 Method

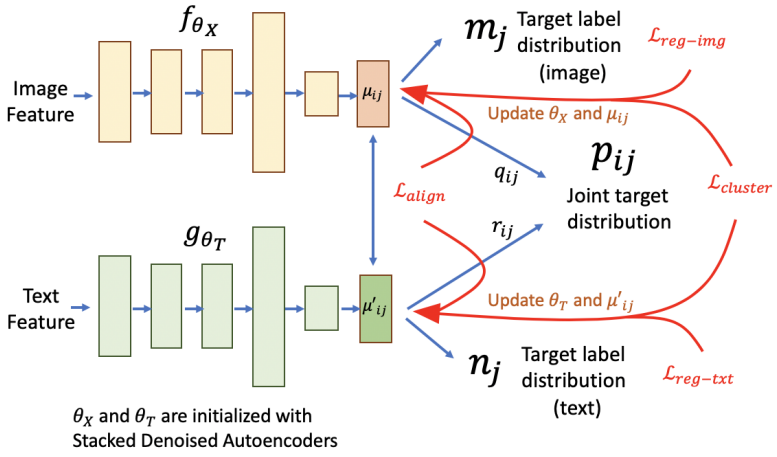


Figure 6.1: Overview of JECL. The initialization phase initializes DNN parameters and centroids using a stacked denoising autoencoder and K-means on the embedded data points. During the clustering phase, parameters and centroids are updated by minimizing the regularized KL divergence between a joint distribution **p** and the image distribution **q** (similarly, text distribution **r**) and the alignment loss between soft cluster assignments of text and images. This figure is best viewed in color.

Figure 6.1 shows an overview of our method. JECL clusters data by simultaneously learning 1) DNN parameters θ_X and θ_T that map each data point with image feature x_i to an embedding $z_i \in Z$ and each text feature t_i to an embedding $z'_i \in Z'$, and 2) set of image cluster centroids μ_j in Z and a set of text cluster centroids μ'_j in Z'.

Parameter Initialization We initialize DNN parameters θ_X and θ_T with two stacked denoising autoencoders. Stacked denoising autoencoders have shown success in generating semantically meaningful representations for both text and images in several studies (c.f., [381, 232, 405]). We train the stacked denoising autoencoders to learn the initial DNN parameters for each view by minimizing mean square error reconstruction loss. After training the autoencoders, we discard the decoders, pass data x_i and t_i through the trained encoders to obtain the initialized embeddings z_i and z'_i. Then, we apply K-means to the embeddings z_i and z'_i to obtain initialized centroid sets μ_j and μ'_j.

Soft Assignment Following Xie et al. [405], we model the probability of data point i being assigned to cluster j using the Student's t-distribution [263], producing a distribution (q_{ij} for images and r_{ij} for text).

$$q_{ij} = \frac{(1 + \left\| z_i - \mu_j \right\|^2 / \alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'} (1 + \left\| z_i - \mu_{j'} \right\|^2 / \alpha)^{\frac{\alpha+1}{2}}} \tag{6.1}$$

$$r_{ij} = \frac{(1 + \left\| z'_i - \mu'_j \right\|^2 / \alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'} (1 + \left\| z'_i - \mu'_{j'} \right\|^2 / \alpha)^{\frac{\alpha+1}{2}}} \tag{6.2}$$

where q_{ij} and r_{ij} are the soft assignments for images and text, respectively, and α is the number of degrees of freedom of the Student's t-distribution.

Cluster Alignment After calculating the soft assignments for both views, we must align the two sets of k clusters, since the j-th image cluster does not necessarily correspond to the j-th text cluster. To achieve this, we use the popular Hungarian algorithm [222] to obtain the alignment between image clusters and text clusters. The Hungarian algorithm is an optimization algorithm to solve the assignment problem by minimizing the assignment

cost. We create a $k \times k$ confusion matrix where an entry (m, n) represents the number of data points being assigned to m-th image cluster and n-th text cluster. We then subtract the maximum value of the matrix from the value of each cell to obtain the "cost." The Hungarian algorithm is then applied to the cost matrix to find a clustering assignment with the minimum cost.

Clustering with KL Divergence Minimization Similar to Xie et al., we refine the cluster centroids by leveraging high-confidence assignments using an auxiliary joint target distribution. However, JECL is trained by matching both the image soft assignments $\mathbf{q}$ and the text soft assignments $\mathbf{r}$ to a *single* target distribution $\mathbf{p}$, which allows information passing between the two models. Specifically, JECL minimizes the KL divergence between $\mathbf{p}$ and $\mathbf{q}$ and the KL divergence between $\mathbf{p}$ and $\mathbf{r}$. The joint loss is as follows:

$$L_{cluster} = \underbrace{KL(\mathbf{p}||\mathbf{q})}_{\text{image loss}} + \underbrace{KL(\mathbf{p}||\mathbf{r})}_{\text{text loss}} \tag{6.3}$$

$$= \frac{1}{N} \sum_{i}^{N} \sum_{j}^{k} \left\{ p_{ij} \log \frac{p_{ij}}{q_{ij}} + p_{ij} \log \frac{p_{ij}}{r_{ij}} \right\} \tag{6.4}$$

Choice of Joint Target Distribution The target distribution $\mathbf{p}$ aims to improve cluster purity and to emphasize data points with high assignment confidence [405]. Our preliminary design adapted this idea to the multi-view problem setting by using a separate target distribution for text and images sub-models separately. But we found that aligning both images and text to the same joint target distribution simplified the model, improved performance, and was more robust to noise. The JECL target distribution is as follows.

$$p_{ij} = \lambda \times \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_{j'} q_{ij'}^2 / \sum_i q_{ij'}} + (1 - \lambda) \times \frac{r_{ij}^2 / \sum_i r_{ij}}{\sum_{j'} r_{ij'}^2 / \sum_i r_{ij'}} \tag{6.5}$$

where λ is a hyperparameter controlling the relative influence of the images and texts, which we set to 0.5 for all experiments but may be optimized or selected for specialized domains. We found that tuning λ had only a mild effect on performance on our datasets in Section 6.5. This model is naturally robust to missing text or missing images: Missing text causes the second term in equation (6.5) to be 0, such that the data points *with* text have a higher

value of p_{ij} and therefore contribute a larger gradient to the model. We will show this effect experimentally in Section 6.5.

Cross-Modality Alignment Loss To better exploit the paired features of our data, we apply cross-modality alignment loss to force the soft assignments of every image-caption pair to be similar. The idea is that the text and image from the same pair should be assigned to the same cluster (and, more generally, should have similar soft assignment distributions). We use Jensen–Shannon divergence (JSD) [343] to capture the similarity between the cluster assignment distributions:

$$L_{align} = JSD(\mathbf{q}||\mathbf{r}) \tag{6.6}$$

$$= \frac{1}{2}KL(\mathbf{r}||\mathbf{s}) + \frac{1}{2}KL(\mathbf{q}||\mathbf{s}) \tag{6.7}$$

where $\mathbf{s} = \frac{1}{2}(\mathbf{q} + \mathbf{r})$. JSD is preferred in this setting over KL divergence because it is symmetric and always has a finite value.

Distribution Regularizer Many clustering algorithms are prone to producing trivial solutions and empty clusters [97, 55]. Distribution regularizers can significantly improve clustering performance [97, 250] in these situations. Dizaji et al. [97] used a regularization term to penalize non-uniform cluster assignments. In our case, the overall distribution of the data is unknown and we can not assume that the distribution should be uniform. Instead, we apply a regularizer on each view to avoid empty clusters and maintain freedom for the overall distribution. We define a target label distribution by averaging the soft frequencies for every view.

$$m_j = q(y = j) = \frac{1}{N}\sum_{i}^{N} r_{ij} \tag{6.8}$$

$$n_j = r(y = j) = \frac{1}{N}\sum_{i}^{N} q_{ij} \tag{6.9}$$

where m_j and n_j can be interpreted as the prior frequency of clusters for image and text, respectively. To impose the preference of a balanced assignment, we add a term representing the KL divergence from a uniform distribution $\mathbf{u}$. The regularized KL divergence is computed

as

$$L_{\text{reg-img}} = KL(\mathbf{m}||\mathbf{u}) \tag{6.10}$$

$$L_{\text{reg-txt}} = KL(\mathbf{n}||\mathbf{u}) \tag{6.11}$$

and the overall regularized term can be summarized as:

$$L_{reg} = L_{reg-img} + L_{reg-txt} \tag{6.12}$$

The overall loss function is as follows.

$$L_{JECL} = L_{cluster} + \gamma L_{align} + \beta L_{reg} \tag{6.13}$$

where the first term aims to minimize the dissimilarity between the soft assignment distribution and the joint target distribution as a clustering objective, the second term is to penalize dissimilar soft assignments from the image and text of the same pair, and the last term is to force the model to prefer a balanced assignment for each view to prohibit empty clusters. We have γ and β as hyperparameters to adjust the weightings of the alignment term and regularized term. We will show that JECL is stable to our hyperparameters in Section 6.5.

Optimization During the training process, we alternate between two steps. In the first step, we compute the target distribution p_{ij} from q_{ij} and r_{ij}. In the second step, we fix p_{ij} to update q_{ij} and r_{ij} by refining the parameters (θ_X and θ_T) and cluster centroids (μ_j and μ'_j) via gradient descent. The process continues until convergence.

Final Cluster Assignment After convergence is met, JECL learns a pair of representations (z_i, z'_i) for each image-text pair (x_i, t_i). The final cluster assignment y_i can be obtained by

$$y_i = \arg\max_j p_{ij} \tag{6.14}$$

6.4 Experiments

We evaluate JECL with four benchmark datasets and compare to both single-view and multi-view algorithms.

Dataset	# Points	# Categories	average # words	% of largest Class	% of smallest Class
Coco-cross	7429	10	50.5	23.2 %	1.6%
Coco-all	23189	43	50.4	7.4%	0.4%
Pascal	1000	20	48.9	5.0%	5.0%
RGB-D	1449	13	38.5	26.4%	1.7%

Table 6.1: Dataset statistics.

6.4.1 Datasets

To evaluate our method, we choose benchmark datasets that have images with corresponding captions as well as ground-truth labels to define the clusters. We summarize the datasets in Table 6.1. (1) **Coco-cross**: MSCOCO [249] is a large-scale object detection, segmentation, and captioning dataset. There are five sentences of captions per image. We discard images containing multiple objects and only consider the largest category from ten supercategories. Finally, we have 7,429 data points from these ten categories in total. (2) **Coco-all**: For this subset of MSCOCO, similar to Coco-cross, we remove images with more than one object, and we remove all categories that include less than 100 images. The result is a dataset with 23,189 images from 43 categories. (3) **Pascal** [321]: This dataset contains 1,000 images with 20 categories, 50 images each category. Every image is associated with five sentences. (4) **RGB-D** [207]: This dataset includes 1,449 images with 13 indoor scenes. Every image is captioned with a paragraph which describes the content of the image. Compared to Coco and Pascal datasets, the captions in this dataset are less specific to the categories and significantly less reliable as a source of information.

6.4.2 Competitive Methods

We compare our method to a variety of single-view and multi-view methods.

Single-View Methods We run two single-view methods to serve as baseline comparisons: **K-means** (KM) [256] and **Deep Embedded Clustering** (DEC) [405].

Multi-View Methods We evaluate six state-of-the-art multi-view methods, including three multi-view representation learning models and three multi-view clustering models. We also evaluate a naive baseline for multi-view methods that simply concatenates the ResNet-50 features and the Doc2Vec features before applying K-means and DEC as in the single-view case. (1) **VSE** [199]: Unifying Visual-Semantic Embeddings unifies joint image-text embedding models by minimizing pairwise ranking loss. K-means is implemented to acquire the cluster centroids. (2) **DCCA** [18]: Deep Canonical Correlation Analysis learns complex nonlinear transformations of two views of data by maximizing the regularized total correlation. The cluster assignments are obtained with K-means on the joint representations. (3) **CCAL-**L_{rank} [101]: This method learns a joint representation by maximizing Canonical Correlation with a pairwise ranking loss. K-means is applied to the learned embeddings. (4) **DMF-MVC** [424]: Multi-View Clustering via Deep Matrix Factorization is a deep matrix factorization framework to learn the semantic information of all views in a layer-wise fashion. (5) **BMVC** [423]: Binary Multi-view Clustering is a joint learning framework simultaneously compressing inputs into collaborative binary representations and clustering the collaborative representations using a binary matrix factorization model. (6) **MLRSSC** [49]: Multi-view Low-rank Sparse Subspace Clustering is a multi-view spectral clustering framework that learns a joint subspace representation by building affinity matrix among all views with the constraints of low-rank and sparsity.

6.4.3 Evaluation Metrics

All experiments are evaluated by three standard clustering metrics: clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI). For all metrics, higher numbers indicates better performance.

6.4.4 Implementation

We use a ResNet-50 model [151], pretrained on the 1.2M images from ImageNet [91], to extract 2048-dimensional images features and Doc2vec [231], pretrained on Wikipedia via skip-gram, to obtain 300-dimensional text features. All methods are fed with these pre-

trained features except for VSE, which has raw text as input. We use hyperparameter settings following Xie et al. [405] for DEC components. λ, γ, and β are set to 0.5, 0.1, and 0.1 in all experiments, respectively. We will show that the model remains stable to different hyperparameter settings within reasonable range in the next section. For baseline algorithms, we use the same setting in their corresponding paper. All the results are the average of 5 trials .

	Coco-cross			Coco-all			Pascal			RGB-D		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Single-View (Image)												
ResNet-50 + KM	0.647	0.712	0.558	0.519	0.614	0.442	0.486	0.516	0.307	0.353	0.289	0.161
ResNet-50 + DEC	0.649	0.629	0.670	0.472	0.701	0.429	0.418	0.564	0.311	0.421	0.352	0.236
Single-View (Text)												
Doc2Vec + KM	0.720	0.852	0.737	0.613	0.807	0.589	**0.544**	0.602	0.398	0.438	0.384	0.279
Doc2Vec + DEC	0.720	0.843	0.729	0.557	0.738	0.501	0.295	0.294	0.120	0.429	0.383	0.287
Concatenation of Both Views + Single-View Models												
Concat(ResNet50+Doc2Vec) + KM	0.636	0.711	0.550	0.517	0.617	0.439	0.478	0.517	0.302	0.355	0.290	0.211
Concat(ResNet50+Doc2Vec) + DEC	0.737	0.758	0.677	0.419	0.550	0.275	0.225	0.326	0.121	0.344	0.255	0.172
Multi-View Representation Learning												
VSE + KM	0.665	0.736	0.607	0.520	0.628	0.430	0.479	0.508	0.300	0.388	0.318	0.194
DCCA + KM	0.712	0.822	0.703	0.645	**0.817**	0.603	0.442	0.485	0.238	0.388	0.310	0.186
CCAL-L_{rank} + KM	0.699	0.806	0.689	0.641	0.812	0.587	0.446	0.489	0.224	0.404	0.316	0.196
Multi-View Clustering												
BMVC	0.365	0.227	0.200	0.410	0.441	0.316	0.392	0.378	0.214	0.207	0.088	0.047
MultiViewLRSSC	0.726	0.781	0.706	0.569	0.747	0.530	0.534	0.574	0.371	0.474	0.400	0.277
DMF-MVC	0.829	0.805	0.774	0.632	0.776	0.608	0.512	0.573	0.380	0.441	0.330	0.257
JECL	**0.929**	**0.908**	**0.934**	**0.675**	0.801	**0.685**	0.512	**0.625**	**0.403**	**0.543**	**0.472**	**0.367**
w/o alignment	0.922	0.906	0.931	0.634	0.784	0.643	0.502	0.613	0.332	0.513	0.423	0.277
w/o regularizer	0.894	0.890	0.889	0.624	0.777	0.610	0.513	0.620	0.376	0.520	0.433	0.327
w/o alignment & regularizers	0.863	0.878	0.852	0.611	0.757	0.607	0.487	0.579	0.352	0.502	0.413	0.367

Table 6.2: Clustering performance of several single-view and multi-view algorithms on four datasets. The results reported are the average of five iterations. JECL outperforms competitive methods on three datasets by a large margin. We also conduct an ablation study on the regularization term and alignment loss. The experimental results show that both additions improve the model significantly.

6.5 Experimental Results

Table 6.2 displays the quantitative results for different methods on various datasets. JECL outperforms other tools on almost every dataset by a significant margin. The table also contains the results of an ablation study of the distribution regularizer and distribution alignment loss. We can see that both additions improve the overall performance.

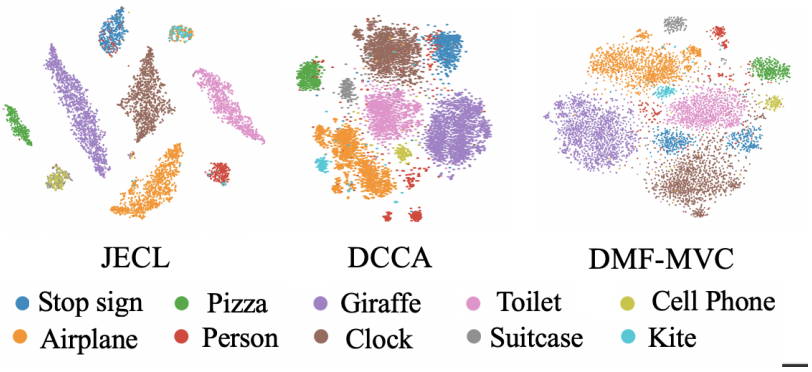


Figure 6.2: Clustering behavior of JECL, DCCA and DMF-MVC. Color indicates ground-truth labels. Cluster shape and position is not meaningful. JECL successfully separates semantically distinct clusters with clear boundaries between clusters. While DCCA and DMF-MVC are able to gather semantically similar images, the boundaries between clusters are unclear, which is reflected in the quantitative performance.

Qualitative Comparison The cluster metrics are difficult to interpret, so we are interested in exploring a qualitative comparison between JECL and the best single-view (DEC), image-text representation learning (DCCA), and multi-view (DMF-MVC) competitors. Figure 6.2 is a visualization of the latent space of JECL to illustrate its effectiveness in producing coherent clusters. We use t-SNE to visualize the embeddings from the latent space. The positions and shapes of the clusters are not meaningful due to the operation of t-SNE. JECL is able to generate semantically distinct clusters with clear boundaries between clusters. While DCCA and DMF-MVC are able to associate semantically similar images, the cluster boundaries are less distinct. We further compare JECL to DEC to examine the effect of

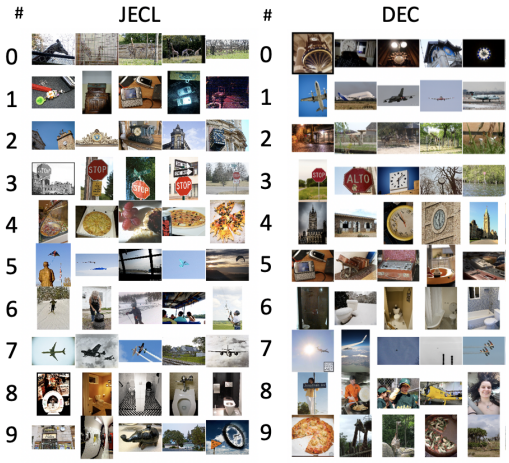


Figure 6.3: The 5 highest-confidence images in each cluster from JECL and DEC. JECL clusters appear qualitatively better. For example, airplanes and kites, two visually and semantically similar concepts, are clearly distinguished, while DEC appears to struggle to distinguish giraffes and pizza.

additional view and our alignment loss by inspecting examples of the clusters. Figure 6.3 shows the top five images with highest confidence from each cluster from the *Coco-cross* dataset. The figure shows that DEC clusters are not always coherent. For example, cluster #1 and cluster #7 seem to include mostly *airplane* images and cluster #0 and cluster #4 are *clock* clusters. Cluster #9 from DEC is a fusion of *giraffe* and *pizza*, which are not at all similar semantically. Our guess is that both *giraffe* and *pizza* share similar colors (yellow) and patterns (spots on the giraffe body and toppings on the pizza). JECL, on the other hand, is easily able to distinguish these objects, because the text descriptions expose their semantic differences. Surprisingly, JECL is also able to distinguish *airplane* and *kite*, which are not only visually similar, but are also semantically related. However, we are still able to observe some errors from JECL, such as examples of *suitcase* and *cellphone*, which are visually similar, assigned into the same cluster (cluster #1) and *clock* examples separated into two clusters: clocks on towers (cluster #2) and indoors clocks (cluster #9). *To summarize, JECL appears to tolerate ambiguity better than other methods.*

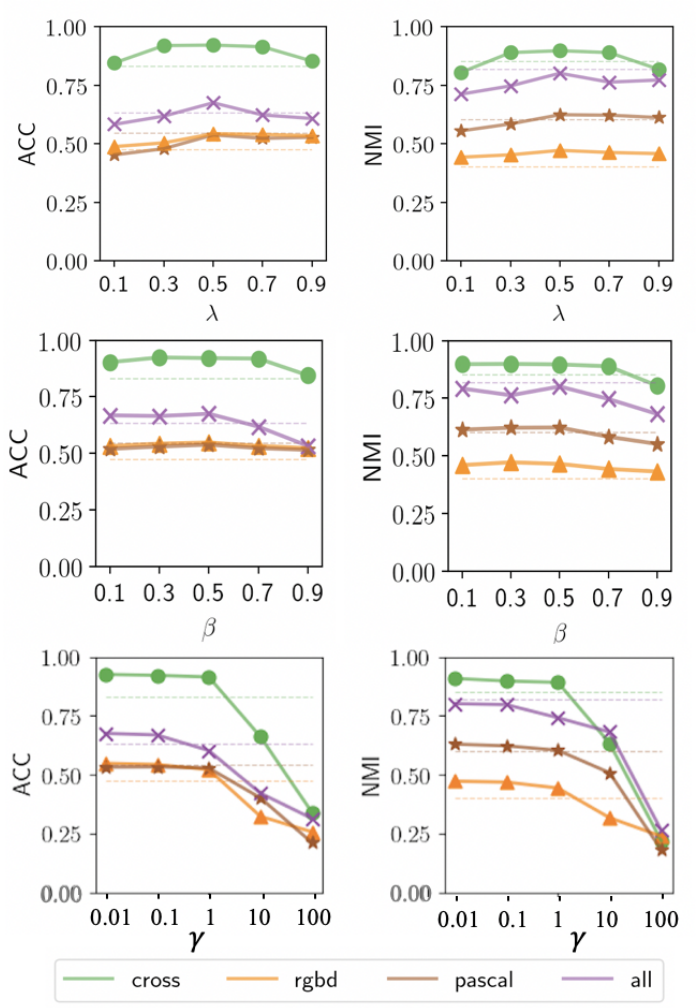
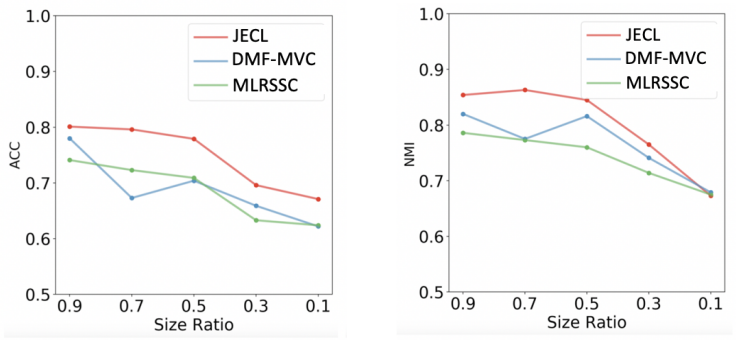


Figure 6.4: The experimental results of hyperparameter sensitivity. The dash lines are the best performing competitive algorithms listed in Table 6.2. JECL is generally robust to hyperparameter settings, while is the most stable and produces top results with $\lambda = 0.5$, $\beta = 0.1$ and $\gamma = 0.1$ among all datasets.

Model Sensitivity to Parameters We have three hyperparameters, λ, β, and γ, to control the weighting of text and image, distribution regularizer, and alignment loss, respectively. JECL is designed for unsupervised learning scenarios, where the training data is limited and hyperparameters tuning is unachievable. In this section, we will show that JECL

is robust to different hyper-parameters settings within reasonable range. Figure 6.4 shows the model performance on various hyperparameter settings. On the weighting between text and image λ, we discover that the model is stable with λ between 0.3 and 0.7 and performs the highest or close to the highest when $\lambda = 0.5$, which means JECL has equal weighting on text and images. This is not surprising. As we mention in Section 6.3, JECL naturally learns from high confidence data points. In DEC, data points close to cluster centroids are those with high confidence which contribute more to the gradients. In multi-view setting, JECL learns from data points with consistent soft assignments of the text and image pairs with the aid of high confidence data points from all views. With this mechanism, tuning λ is trivial and we will demonstrate it in missing view experiment. Then, we study the effect of hyperparameter for distribution regularizer term, β. From Figure 6.4, we can observe that the model performance remains stable but slowly deteriorates when β is close to 1. The number of empty clusters decreases by 23% when the regularizer is applied. Finally, we investigate the influence on the hyperparameter γ, which is used to adjust the text-image alignment loss. We use a wider range to test the γ, because we wonder whether a stronger image-text bond would help overall clustering performance. We can see that the overall model performance remains steady with $\gamma < 1$ and it drops dramatically with $\gamma > 1$. The reason for this deterioration is that the alignment loss dominates overall loss and causes the clustering to perform poorly. *To summarize, JECL is robust when β and γ are smaller than 1.*

Sensitivity to Data Size We consider JECL performance with varying data size. In order to evaluate the performance of JECL on varying data size, we produce a subset of *Coco-all* (labeled *all-sub* in the figures and tables) with 8000 image-text pairs: 8 classes with 1000 image-text pairs each. We then sample this subset to vary the data size. We compare against the two state-of-the-art multi-view clustering methods, DMF-MVC and MLRSSC. Figure 6.5 shows that JECL's performance is robust until only 500 data points in each class remain and drops dramatically when only 100 data points in each class remain. Despite this drop in performance, *JECL continues to outperform competitive models, DMF-MVC and MLRSSC on these smaller data sizes.*



(a) (b)

Figure 6.5: JECL performance as data size decreases. The performance degrades when size ratio is below 0.5 (500 data points in each class), while JECL still outperforms the state-of-the-art multi-view clustering methods, DMF-MVC and MLRSSC on varying data sizes.

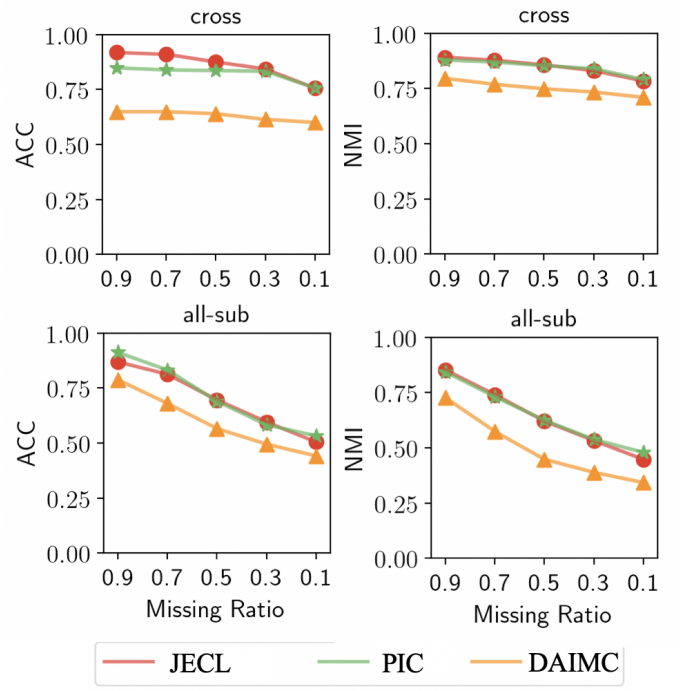


Figure 6.6: Experimental results on missing view scenarios. JECL is competitive with the state-of-the-art method, PIC, and outperforms DAIMC by a large margin on both datasets.

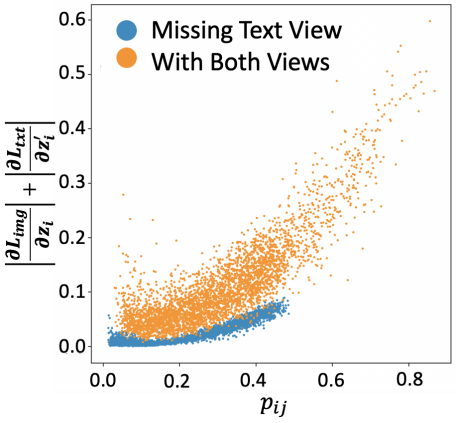


Figure 6.7: JECL's robustness to missing data is attributable to the model of the joint distribution: the images with text (orange) contribute more to the gradient than the images with missing text (blue).

Robustness to Missing Text Incomplete views are a common problem in multi-view clustering [408]; we cannot expect all images to be equipped with text descriptions. To analyze the robustness of JECL when text descriptions are missing, we remove text from a random set of images at varying rates on *Coco-cross* and *all-sub* dataset. We compare our method against two state-of-the-art incomplete multi-view clustering model, PIC [385] and DAIMC [168]. The missing view experimental results appear in Figure 6.6. JECL is competitive with PIC on both datasets and outperforms DAIMC by a large margin. Figure 6.7 demonstrates that images with captions (orange dots) have larger value of p_{ij} and contribute a larger gradient to the training process. Images with missing text have a smaller associated value of p_{ij} because the second term in equation (6.5) vanishes.

6.6 GraviTIE

Countless images exist on the web. These include photographs, memes, art images, scientific figures, and more. Analysis of these image datasets increasingly requires the application of large-scale machine learning methods to answer even relatively straightforward questions.

For example, the dataset recently released by Twitter[1] contains rich information about Russian influence, but even conceptually simple questions such as "Do images posted from Russian accounts tend to differ from those posted by other accounts?" require a combination of unsupervised learning (to determine image similarity), NLP methods (to contextualize the image), and structured queries (to compare images by source and year).

These analysis tasks have several defining characteristics. First, tasks increasingly require some form of unsupervised learning. Ground truth labels or other obvious sources of supervision are rarely available. In these situations, interactive, qualitative exploration of the results is the only option. Second, the data is *multi-modal*: each record consists of some combination of an image, free text, and a set of structured attributes. Each of these elements may be of arbitrary size, or altogether missing. Third, the datasets are large, making direct-browsing and main-memory approaches infeasible. For example, the Twitter dataset contains 9 million tweets with over 543k images, the Artstor dataset contains more than 2 million high-quality images of artwork along with text descriptions, and the Viziometrics [237] dataset consists of over 9 million figures from the scientific literature, along with their captions.

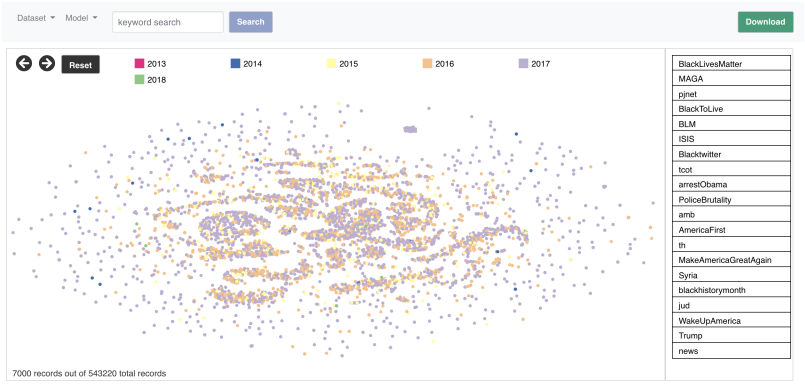


Figure 6.8: Main interface of GraviTIE. GraviTIE (Global Representation and Visualization of Text and Image Embedding) is an interactive web visualization application and discovery engine for large image-text datasets.

[1] https://about.twitter.com/en_us/values/elections-integrity.html#data

In this demonstration, we present GraviTIE (**G**lobal **R**epresentation **a**nd **Vi**sualization of **T**ext and **I**mage **E**mbeddings, pronounced "gravity"), an interactive visualization system for large-scale image-text datasets. Fig. 6.8 shows the interface of GraviTIE.

For pre-processing, GraviTIE takes as input a large set of image-text pairs. It learns an embedding for the pair with clustering objective by running JECL on the text and image, such that the two models tend to produce the same distribution. The properties of MultiDEC are tightly coupled to the design of the interface. In particular, this model 1) accommodates multi-modal heterogeneous data while 2) maintaining reasonable performance at scale. The learned embedding is then dimension-reduced to two dimensions for display purposes.

The GraviTIE interface features a large similarity map to display thousands of nodes, where each of the node represents an image positioned by a multi-modal vector embedding. Similar images are clustered together on this similarity map which helps the users efficiently navigate a large number of images. Users can refine the current view using a combination of free-text queries (with appropriate operators “must include”, “might contain” or “exclude”) and structured query filters. Only a sample of the images is displayed to ensure performance at scale and to afford hover interactions with individual images; displaying all images would lead to extremely dense regions with significant occlusion. Users can pan, zoom, filter, and search iteratively, gradually refining the set of displayed images and inspecting the images in each cluster via hover interactions at any time. Each view is associated with a unique url (i.e. RESTful), making the system stateless and programmable, allowing refinements to be saved and shared with collaborators. The data displayed in each view can be directly downloaded to provide custom curated subsets for specific tasks. For example, an art historian can construct a dataset of impressionist paintings by selecting particular clusters and filtering for a particular time period. The RESTful API design also affords additional downstream applications to be developed using the search and download functions.

GraviTIE is designed to be adaptable to a number of applications across multiple modalities and communities. In its current form, art historians are using GraviTIE to explore influence patterns in art, computer scientists are using it to qualitatively assess unsupervised learning methods, and social scientists are using it to understand the behavior of Russian actors on Twitter.

6.6.1 Background

Analysis and visualization of large scale images have been receiving great attention from researchers. Hochman et al. [160] explore over 2 million photos on Instagram by sorting the images in terms of their properties (e.g., hue or create time) and showing the complete image sets of different cities to reveal the local culture and social patterns. The T-SNE Map [2], one of Google Arts & Culture experiments by Diagne et al., is a 3D interactive platform created by computing the visual similarity of art works and grouping the art works with t-SNE algorithm. PixPlot[3] extends the idea by applying UMAP [274] algorithm which can scale large data set and cluster million of data points and allows users to implement custom datasets. These applications provide a simple platform to explore large scale image sets, but do not support query and do not exploit text or structured metadata to assist search and curation. Wang et al. [384] introduced iMap, a treemap-based visualization for navigating image search and clustering results, but the number of images displayed is limited. iGraph [136] constructs a similarity graph by computing the affinity between images, the relationship between texts, and the connection between images and texts, which results in a system that requires enormous amount of computation.

6.6.2 GraviTIE System Overview

Figure 6.9 summarizes the architecture of GraviTIE. The pipeline can roughly be divided into three parts: Constructing similarity map, textual and numeric data storage, and image storage.

Learning the Similarity Map: The front end interface of GraviTIE takes a 2-d vector of coordinates for each image-text pair to construct the similarity map visualization. In this demonstration, all datasets are pre-processed by the same pipeline in which we first separately embed image and text, acquire a combined representation vector for each text and image pair, and finally apply a dimensional reduction algorithm to obtain the desired 2-d numeric vectors. We use MultiDEC [414] to learn representations of these image-caption

[2]https://artsexperiments.withgoogle.com/tsnemap/

[3]http://dhlab.yale.edu/projects/pixplot/

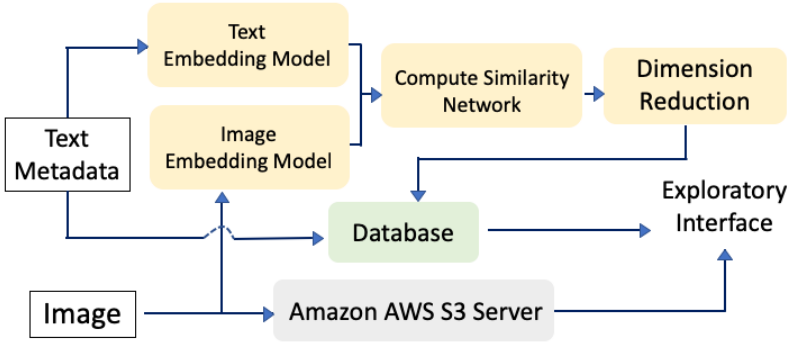


Figure 6.9: GraviTIE system overview. Yellow: learning the similarity map. Green: learned embeddings and associated metadata are stored in a database. Gray: images are stored in a cloud-based object store.

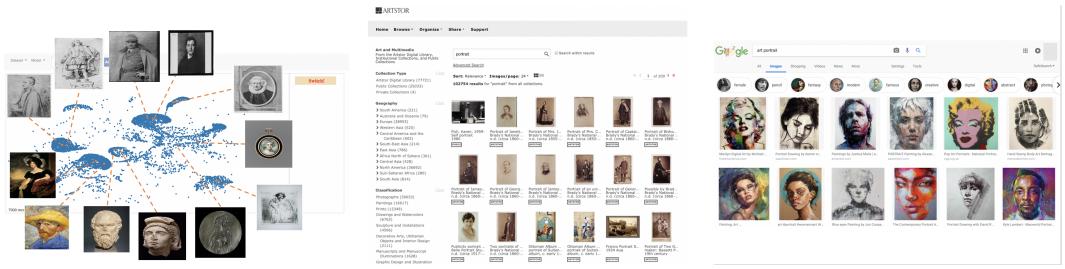
pairs. MultiDEC is a method for clustering image-caption pairs by iteratively computing an auxiliary target distribution and matching both image distribution and text distribution to the target. Because MultiDEC produces distinct and semantically meaningful representations for both image and text, it is a natural candidate for this task. Users can also visualize the visual features and textual features alone.

Textual and Numeric Data Storage: The metadata, textual description, and the 2-d numeric vector for each image-text pair is saved in a MySQL database. GraviTIE also utilizes MySQL full text search (boolean mode) to support keyword search.

Image Storage: GraviTIE uses Amazon AWS S3 to store millions of image for display purposes.

6.6.3 GraviTIE Demonstration Datasets

We will demonstrate the GraviTIE system on three large-scale datasets to show the versatility and the breadth of GraviTIE.



(a) Search for "portrait" on GraviTIE. (b) Search for "portrait" on Artstor. (c) Search for "portrait" on Google Images.

Figure 6.10: Search results for "portrait" using (a) GraviTIE, (b) Artstor, and (c) Google Images. GraviTIE summarizes the space of relevant images; different clusters represent different types of portraits. Artstor and Google Images each show only a small set of top-ranked images, and each engine uses a different opaque ranking function, making the results unpredictable.

Russian Tweets Twitter [4] released over 9 millions tweets from 3,841 accounts that are believed to be connected to the Russian Internet Research Agency. This dataset has received significant attention from researchers to understand foreign influence on US politics. We process and display all 543,220 tweets with images on GraviTIE to investigate relationships from a visual perspective.

In this particular demonstration, we first embed images with ResNet-50 [151] model, pre-trained on 1 million ImageNet dataset[91], and acquire a text representation with a Glove [311] model that is pre-trained on 2 billion tweets. We then apply *multiDEC* [415] to learn a joint representation for the combined image-text pair. Finally, UMAP [274] is performed to produce 2-d vectors for visualization.

Art Imagery GraviTIE includes 2,144,301 art images from the ArtStor collection. Artstor is a non-profit organization that creates digital libraries for scholars arts, humanities, social sciences and architecture. They approached us looking for new ways to explore their large

[4] https://about.twitter.com/en_us/values/elections-integrity.html#data

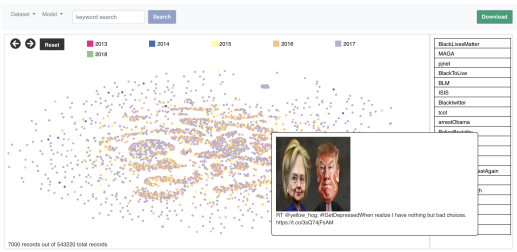
image sets that went beyond simple metadata queries. For these images, the metadata is sparse, so the results of the queries are unreliable. A goal of GraviTIE is to utilize the full coverage of arbitrary data, with no assumptions on completeness. We also want to visually convey the results of unsupervised image clustering. A group of art historians are currently using GraviTIE to find patterns of influence and providing us with feedback.

We extracted visual features for the art images using pre-trained ResNet-50. We concatenate values of title, art classification, creator, repository, and date to serve as the description of each image. We then trained a FastText [41] model based on the description and acquired the textual/meta-data representation for each art work. Similar to the pipeline for the twitter data, we applied *multiDEC* followed by *UMAP* to obtain the final 2-d representation.

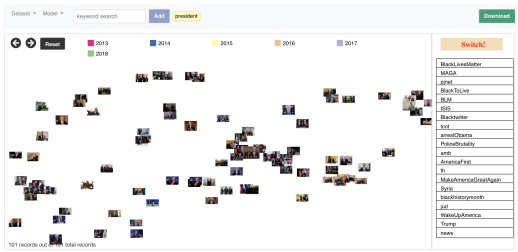
Scientific Figures In addition to the art images and twitter data sets, we use a large database of scientific figures that we have been extracted from PubMed, arXiv and other repositories of the scientific literature. The goal is to understand how the use of different types of figures vary over time and across fields. The example dataset includes over 5 million figures from 800k scientific papers from the arXiv.org. ArXiv provides the raw pdfs; we extract images and captions from these pdfs using pdffigures2.0 [71] and then follow the similarity map pipeline described previously.

6.6.4 GraviTIE Key Features

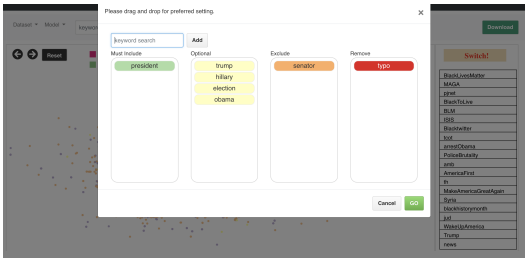
GraviTIE supports interactive queries and the exploration images *collections*. The central feature is a similarity map supporting mixed queries, panning, zooming and visual inspection of individual images. Each image-text pair is represented as a single glyph colored by a user-configurable attribute and situated in the 2-D plane based on the similarity with other image-text pairs. Users can hover over each glyph to view the corresponding image and text content associated with that image (Fig. 6.11a). The query system (Fig. 6.11c) affords dataset refinement and curation via keyword search or structured predicates. Users can customize the current view through a highlighting feature (Fig. 6.11d). These customizations can easily be saved and shared with other users since each view is associated with a unique url. Below we highlight other features of the system:



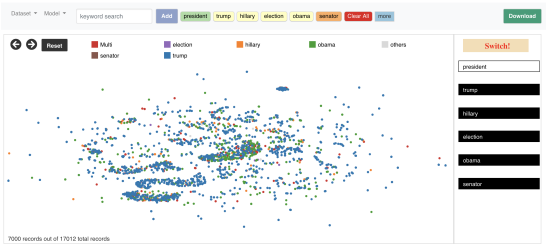
(a) Similarity Map: Each point represents an image-text pair. The user can hover on a point to display the visual and textual content.



(b) Direct inspection: When the number of records is less than 300, thumbnail images are displayed instead of glyphs.



(c) Advanced search: Keyword semantics are configurable.



(d) Highlighting: Items with a selected keyword are highlighted.

Figure 6.11: Some features of GraviTIE: (a) Similarity map, (b) Direct inspection, (c) Advanced search, (d) Highlighting

Similarity Map : The similarity map is what distinguishes the design of GraviTIE from other systems for interacting with large image collections. It makes no assumptions about the images (unlike, say Building Rome in A Day [9]), and can make use of multi-modal information if available. The similarity map can also be used as a replacement for conventional list-based search interfaces: by providing a visual summary of all results rather than the top few items in a ranked list, we allow the user to process information and identify patterns faster than scanning a list of textual information [292].

Consider the simple experiment in Figure 6.10. We search for the keyword *portrait* on three platforms: GraviTIE, Artstor, and Google Images. GraviTIE reveals that there are

several dense clusters of similar images, and upon inspection, these clusters are easily identified as corresponding to print portraits, painted portraits, photograph portraits, and other semantic types. However, the Artstor search results display only the top k images ranked by query relevance, which in this case consist entirely of print portraits. The ranking function used by Google Images makes another seemingly arbitrary choice, displaying entirely painted portraits. Not only do both of these sets of results seem incorrect in general, but even for those situations where they are what the user wants, GraviTIE can emulate similar results within a few hover-and-zoom interactions.

Scalability, Query, and Reproducibility : The similarity map also directly affords scalability. By simply applying a sampling mechanism, we can retain the overall structure of the (embedded) image space while controlling system performance and not occluding individual images. By combining offline pre-processing and sampling, we maintain high performance interactions for datasets of arbitrary scale. Cluster-aware sampling approaches and their impact on interactivity and user task performance remain future work.

GraviTIE supports exploring a set of related keywords at a time. Given a set of user-provided search terms, the user can force each term to be included or excluded. For instance, the results of the query result of Fig.6.11c would definitely include the term *president*, might contain *trump*, *hilary*, *election*, and *obama*, and would exclude *senator*. In addition, selecting a keyword highlights the elements in the similarity map associated with that keyword, as in Fig. 6.11d. To initialize this interaction, users can toggle between default interesting keywords (eg. Fig. 6.11b shows the top 20 frequent hashtags) or type their own (eg. Fig. 6.11d). If an item includes multiple highlighted keywords, it is indicated with a distinguished color.

Every view in GraviTIE is associated with a unique url to afford collaboration, provenance, and sharing. The browser history can be used to retrace steps, and particular views can be saved simply by copying and pasting links. In addition, a user can download the corresponding dataset associated with any view for further analysis. This feature has been valuable in supporting research by providing an easy way to construct high-quality training datasets from important subsets of images. For example, GraviTIE can be used to

quickly find a large set of phylogenetic tree images in the scientific literature for information extraction research[240].

6.7 Summary

In this chapter, JECL is proposed to take an image-text pair and simultaneously learn cluster centroids and representations. JECL consists of two parallel encoders with a clustering layer. It is optimized by a clustering loss, cross-modality loss, and a regularizing loss. JECL outperforms current multi-view clustering algorithms in four benchmark datasets and the learned representations present tight clusters and clear boundaries between clusters. GraviTIE is also introduced to accompany JECL for large-scale image-text collections. GraviTIE features a similarity map on which the users can hover, pan, zoom, query, and visually inspect individual images. GraviTIE also affords scalability, query, and reproducibility. In high-expertise applications, while structured image data is hard to acquire, images are often equipped with descriptions. With GraviTIE and JECL, the users are able to navigate through large-scale image collections efficiently.

Chapter 7

CONCLUSION

In this dissertation, we address challenges organizing, discovering, and communicating knowledge resulting from reduced accessibility to large-scale labeled data in machine learning applications within high expertise domains. Our research objectives consider novel solutions to facilitate scientific communication, leverage ontologies in high expertise domains with hierarchical multi-label classification, and model large scale heterogeneous collections of images with short text descriptions. We summarize this dissertation's main contributions and directions future research in this chapter.

Considering novel solutions to facilitate scientific communication: We use machine learning to understand how scientists use figures and references to effectively and efficiently communicate with each other. We show that there are distinctive patterns of figures across sub-disciplines and that these visual patterns expose new modalities of communications which are not identifiable by either text or citation graphs. Fine-grained figures, such as neural network architectures, can also serve as a marker to track propagation and popularity of the ideas and methods. Next, we propose the idea of "central figures". Central figures summarize the main contribution of publications. We show that the authors from 87% of surveyed papers agree central figures exist in their publications, and we present two models to predict the central figure in a research paper. Finally, we demonstrate that references have a surprisingly high predictive power for paper acceptance and that this indicates referencing plays an important role for scientific communication. We believe central figures can contribute greatly in the evolution of scientific communication with quick idea transferring and flexible publishing platforms and that incorporating referencing evaluation in scientific publications will alleviate the burden of the reviewers.

It is unknown if these findings are generalizable due to limited accessibility to structured

data. To be more specific, the survey conducted to verify the existence of central figures is based on publications from PubMed, which focuses on biomedical studies. Do other disciplines, such as physics, engineering, or economics, have similar pattern in visual literature? Moreover, the findings of the predictive power of referencing rely on historical acceptance decisions in the International Conference on Learning Representations (ICLR). Although we provide evidence supporting generalization arguments to broader engineering and economics papers on Arxiv.org, the transferability to biomedical publications is yet to be supported. While we believe that the results of these studies can be applied to other disciplines, we do not know to what extent they can be generalized. Future works should be conducted to study the generalilzation of these findings.

Leveraging ontologies in high expertise applications with hierarchical multi-label classification While large-scale labeled data is uncommon in high expertise domains, human attention tends to invest in curating ontology to describe complex relationships. Although these relationships between terms provide some supervision, this additional information is often ignored by learning solutions. Hierarchical multi-label classification aims to address this problem. In this chapter, we propose Surj, which learns a representation of the label hierarchy and separately learns to map input records onto the learned representation space of the label hierarchy and produces predictions. This framework reduces dependence on the training data and can make predictions even for underrepresented labels. We then introduce Global Hierarchy Violation to measure the hierarchy violations of the predicted outputs. We demonstrate that Surj outperforms existing HMC models by a large margin on 17 of 20 benchmarks. We also discover that the current HMC benchmarks suffer from several deficiencies: (i) A high percentage of labels have little data to train. The HMC problem is arguably harder than the few-shot problem. (ii) The datasets are outdated and lack context. (iii) Although there are 20 common used benchmarks, 16 of them share two label hierarchies. It raises concerns for generalizability. In response, we propose Ontologue, a declarative query system for generating custom benchmarks with specific properties. We then use this system to design four new benchmarks extracted from DBPedia that better represent the problem space. We hope our contributions in a novel model, a metric to

measure hierarchy violations, and new benchmarks can each support and facilitate HMC research.

Heavy resources have been invested in curating ontologies, but there are still many open problems in the context of machine learning research, mainly focused on specialization and fast-changing ontologies. Existing HMC methods struggle to classify unseen labels and often assign a mandatory leaf label even with low confidence. Learning to un-specify is an open issue in HMC problems and it will improve the precision of current HMC solutions. Moreover, ontologies are changing rapidly. Gene Ontology and DBPedia, for example, update their ontologies and annotations every month. The models trained on datasets with old ontologies can become obsolete quickly. Investing in solutions to handle evolving ontologies and the transferability of the trained HMC models to these new ontologies are essential in this space. Another open question that could potentially be answered with the accessibility to large scale data is, "Do these human curated ontologies properly describe the world?" When there are mismatches between data driven ontologies and human-derived ontologies, which one is correct? The answers to these questions can be beneficial to understanding the limitations and utility of the existing ontologies.

Modeling large scale heterogeneous collections of images with short text descriptions: JECL is proposed to take an image-text pair and simultaneously learn cluster centroids and representations. JECL consists of two parallel encoders with a clustering layer. It is optimized by a clustering loss, cross-modality loss, and a regularizing loss. JECL outperforms current multi-view clustering algorithms in four benchmark datasets and the learned representations present tight clusters and clear boundaries between clusters. GraviTIE is also introduced to accompany JECL for large-scale image-text collections. GraviTIE features a similarity map on which the users can hover, pan, zoom, query, and visually inspect individual images. GraviTIE also affords scalability, query, and reproducibility. In high-expertise applications, while structured image data is hard to acquire, images are often equipped with descriptions. With GraviTIE and JECL, the users are able to navigate through large-scale image collections efficiently.

While the empirical evaluation of JECL focuses on image-text pairs, JECL should nat-

urally be applied to any multi-view scenarios. It should also be able to scale to more than two views without any significant changes in the framework. Understanding the generazliation to different domains is an interesting direction for JECL. Consider scientific documents which include several elements and modalities. JECL can take all the elements and learn a joint representation of a scientific document. This can be beneficial to a wide variety of downstream applications.

More and more activities in high expertise domains are moving online [111, 242] beginning with the proliferation of the internet and accelerating over the course of the pandemic. Even after the pandemic, these activities seem to stay online [242]. These digital footprints are producing a lot of messy data and digital exhaust. Taking advantage of these digital footprints is a timely issue. Additionally, these data are increasingly accessible to the general public. For example, `Openreview.net` [352] has hosted hundreds of venues for their peer review process. There are tens of millions of free-access scientific publications in a wide range of disciplines on `Pubmed.gov` and `arXiv.org`. Protein Data Bank [28] has annotations for almost 200k biological macromolecular structures that enable advanced research in the protein folding problem.

With increasing access to these large-scaled high expertise data, I believe unsupervised data curation for high expertise settings will be a crucial area of research. More specifically, we will curate data unsupervisedly, such as AI-generated ontologies. How do the patterns learned from DL differ from human perceptions? How much trust do we have in AI-generated data curation within high expertise domains? I also believe that AI and DL beating human experts will become increasingly common. How we can control AI and use AI to further advance science will be an essential research area.

This dissertation provides an in-depth analyses of current issues in high-expertise domains. Building of these analyses, it makes methodological contributions to machine learning research. It offers novel solutions to promote better scientific communication. The presented analyses provide strategies for researchers and experts to communicate ideas more efficiently. The proposed methods will allow them to organize knowledge and to explore larges-scale unstructured image collections easily. We hope the contributions made in this dissertation will advance machine learning research and facilitate broader scientific studies.

BIBLIOGRAPHY

[1] *SIGCOMM Comput. Commun. Rev. 13-14*, 5-1 (1984).

[2] *CHI '08: CHI '08 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2008), ACM. General Chair-Czerwinski, Mary and General Chair-Lund, Arnie and Program Chair-Tan, Desney.

[3] Frontiers science news, Dec 2018.

[4] Abid, T., Zarzour, H., Laouar, M. R., and Khadir, M. T. Towards a smart city ontology. In *AICCSA 2016* (2016), IEEE, pp. 1–6.

[5] Ablamowicz, R., and Fauser, B. Clifford: a maple 11 package for clifford algebra computations, version 11, 2007.

[6] Abril, P. S., and Plant, R. The patent holder's dilemma: Buy, sell, or troll? *Communications of the ACM 50*, 1 (Jan. 2007), 36–44.

[7] Accomazzi, A., Eichhorn, G., Kurtz, M. J., Grant, C. S., Henneken, E., Demleitner, M., Thompson, D., Bohlen, E., and Murray, S. S. Creation and use of citations in the ads. *arXiv preprint cs/0610011* (2006).

[8] Adya, A., Bahl, P., Padhye, J., A.Wolman, and Zhou, L. A multi-radio unification protocol for IEEE 802.11 wireless networks. In *Proceedings of the IEEE 1st International Conference on Broadnets Networks (BroadNets'04)* (Los Alamitos, CA, 2004), IEEE, pp. 210–217.

[9] Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R. Building rome in a day. In *ICCV* (2009), IEEE, pp. 72–79.

[10] Akker, B. v. d., Markov, I., and de Rijke, M. Vitor: Learning to rank webpages based on visual features. *arXiv preprint arXiv:1903.02939* (2019).

[11] Akyildiz, I. F., Melodia, T., and Chowdhury, K. R. A survey on wireless multimedia sensor networks. *Computer Netw. 51*, 4 (2007), 921–960.

[12] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. Wireless sensor networks: A survey. *Comm. ACM 38*, 4 (2002), 393–422.

[13] Al-Zaidy, R. A., and Giles, C. L. Automatic extraction of data from bar charts. In *K-CAP* (2015), ACM, p. 30.

[14] Al-Zaidy, R. A., and Giles, C. L. A machine learning approach for semantic structuring of scientific charts in scholarly documents. In *AAAI* (2017), pp. 4644–4649.

[15] American Mathematical Society. *Using the amsthm Package*, April 2015. http://www.ctan.org/pkg/amsthm.

[16] Anand, S. S., Bell, D. A., and Hughes, J. G. The role of domain knowledge in data mining. In *Proceedings of the fourth international conference on Information and knowledge management* (1995), pp. 37–43.

[17] Andler, S. Predicate path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN symposium on Principles of Programming Languages* (New York, NY, 1979), POPL '79, ACM Press, pp. 226–236.

[18] Andrew, G., Arora, R., Bilmes, J., and Livescu, K. Deep canonical correlation analysis. In *ICML* (2013).

[19] Anisi, D. A. Optimal motion control of a ground vehicle. Master's thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2003.

[20] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. Vqa: Visual question answering. In *ICCV* (2015).

[21] Archer, Jr., J. E., Conway, R., and Schneider, F. B. User recovery and reversal in interactive systems. *ACM Trans. Program. Lang. Syst. 6*, 1 (Jan. 1984), 1–19.

[22] Arora, S., Liang, Y., and Ma, T. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR* (2016).

[23] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. Gene ontology: tool for the unification of biology. *Nature genetics 25*, 1 (2000), 25–29.

[24] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 2007, pp. 722–735.

[25] Bahl, P., Chancre, R., and Dungeon, J. SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceeding of the 10th International Conference on Mobile Computing and Networking (MobiCom'04)* (New York, NY, 2004), ACM, pp. 112–117.

[26] Bai, X., Zhang, F., and Lee, I. Predicting the citations of scholarly paper. *Journal of Informetrics 13*, 1 (2019), 407–418.

[27] Ban, K.-m. Sustainable development goals.

[28] Bank, P. D. Protein data bank. *Nature New Biol 233* (1971), 223.

[29] Bavoil, L., Callahan, S., Crossno, P., Freire, J., Scheidegger, C., Silva, C., and Vo, H. Vistrails: Enabling interactive multiple-view visualizations. In *IEEE Visualization* (2005).

[30] Beel, J., Gipp, B., Langer, S., and Breitinger, C. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries 17*, 4 (Nov 2016), 305–338.

[31] Bellman, R. *Adaptive control process: a guided tour*. 1961.

[32] Bellman, R. E. *Adaptive control processes: a guided tour*, vol. 2045. Princeton university press, 1961.

[33] Bergstrom, C. T., West, J. D., and Wiseman, M. A. The eigenfactor™ metrics. *Journal of neuroscience 28*, 45 (2008), 11433–11434.

[34] Bertinetto, L., Henriques, J. F., Torr, P. H., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136* (2018).

[35] Bi, W., and Kwok, J. T. Multi-label classification on tree-and dag-structured hierarchies. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (2011), pp. 17–24.

[36] Bi, W., and Kwok, J. T. Multilabel classification on tree-and dag-structured hierarchies. In *ICML 2011* (2011).

[37] Bi, W., and Kwok, J. T. Mandatory leaf node prediction in hierarchical multilabel classification. In *Advances in Neural Information Processing Systems* (2012), pp. 153–161.

[38] Björk, B.-C., and Hedlund, T. A formalised model of the scientific publication process. *Online information review* (2004).

[39] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. Complex networks: Structure and dynamics. *Physics reports 424*, 4-5 (2006), 175–308.

[40] Bohannon, J. Who's afraid of peer review? *Science 342*, 6154 (2013).

[41] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* (2016).

[42] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5* (2017), 135–146.

[43] Boldt, A. Extending arxiv. org to achieve open peer review and publishing. *Journal of Scholarly Publishing 42*, 2 (2011), 238–242.

[44] Borges, H. B., and Nievola, J. C. Multi-label hierarchical classification using a competitive neural network for protein function prediction. In *IJCNN 2012* (2012), IEEE, pp. 1–8.

[45] Bornmann, L., and Mutz, R. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology 66*, 11 (2015), 2215–2222.

[46] Bowman, M., Debray, S. K., and Peterson, L. L. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst. 15*, 5 (November 1993), 795–825.

[47] Boyack, K. W., van Eck, N. J., Colavizza, G., and Waltman, L. Characterizing in-text citations in scientific articles: A large-scale analysis. *Journal of Informetrics 12*, 1 (2018), 59–73.

[48] Braams, J. Babel, a multilingual style-option system for use with latex's standard document styles. *TUGboat 12*, 2 (June 1991), 291–301.

[49] Brbić, M., and Kopriva, I. Multi-view low-rank sparse subspace clustering. *Pattern Recognition 73* (2018), 247–258.

[50] Burbea, J., and Rao, C. On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory 28*, 3 (1982), 489–495.

[51] Burley, R., Moylan, E., Chadwick, C., Newton, A., Preston, A., and Culley, T. What might peer review look like in 2030, 2017.

[52] Buss, J. F., Rosenberg, A. L., and Knott, J. D. Vertex types in book-embeddings. Tech. rep., Amherst, MA, USA, 1987.

[53] Buss, J. F., Rosenberg, A. L., and Knott, J. D. Vertex types in book-embeddings. Tech. rep., Amherst, MA, USA, 1987.

[54] Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. Learning to rank: from pairwise approach to listwise approach. In *ICML* (2007), ACM, pp. 129–136.

[55] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *ECCV* (2018).

[56] Carvalho, M., Cadène, R., Picard, D., Soulier, L., Thome, N., and Cord, M. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In *SIGIR* (2018).

[57] Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. Universal sentence encoder for english. In *EMNLP Demo* (2018), pp. 169–174.

[58] Cerri, R., Barros, R. C., and de Carvalho, A. C. Hierarchical multi-label classification for protein function prediction: A local approach based on neural networks. In *ISDA 2011* (2011), IEEE, pp. 337–343.

[59] Cerri, R., Barros, R. C., and De Carvalho, A. C. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences 80*, 1 (2014), 39–56.

[60] Cerri, R., Barros, R. C., de Carvalho, A. C., and Jin, Y. Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC bioinformatics 17*, 1 (2016), 373.

[61] Cerri, R., Basgalupp, M. P., Barros, R. C., and de Carvalho, A. C. Inducing hierarchical multi-label classification rules with genetic algorithms. *Applied Soft Computing 77* (2019), 584–604.

[62] Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. Incremental algorithms for hierarchical classification. *The Journal of Machine Learning Research 7* (2006), 31–54.

[63] Charbonnier, J., Sohmen, L., Rothman, J., Rohden, B., and Wartena, C. Noa: A search engine for reusable scientific images beyond the life sciences. In *ECIR* (2018), Springer, pp. 797–800.

[64] Charuvaka, A., and Rangwala, H. Hiercost: Improving large scale hierarchical classification with cost sensitive learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2015), Springer, pp. 675–690.

[65] Chatterjee, S., Maheshwari, A., Ramakrishnan, G., and Jagaralpudi, S. N. Joint learning of hyperbolic label embeddings for hierarchical multi-label classification. In *EACL 2021* (2021), pp. 2829–2841.

[66] Chen, B., Huang, X., Xiao, L., Cai, Z., and Jing, L. Hyperbolic interaction model for hierarchical multi-label classification. *arXiv preprint arXiv:1905.10802* (2019).

[67] Chen, B., Huang, X., Xiao, L., Cai, Z., and Jing, L. Hyperbolic interaction model for hierarchical multi-label classification. In *AAAI 2020* (2020), vol. 34, pp. 7496–7503.

[68] Chen, Z., Cafarella, M., and Adar, E. Diagramflyer: A search engine for data-driven diagrams. In *The Web Conference* (2015), ACM, pp. 183–186.

[69] Chollet, F., et al. Keras. https://github.com/fchollet/keras, 2015.

[70] Clare, A. *Machine learning and data mining for yeast functional genomics*. PhD thesis, Citeseer, 2003.

[71] Clark, C., and Divvala, S. Pdffigures 2.0: Mining figures from research papers. In *JCDL* (2016), IEEE, pp. 143–152.

[72] Clark, C., and Divvala, S. Pdffigures 2.0: Mining figures from research papers.

[73] Clark, M. Post congress tristesse. In *TeX90 Conference Proceedings* (March 1991), TeX Users Group, pp. 84–89.

[74] Clarkson, K. L. *Algorithms for Closest-Point Problems (Computational Geometry)*. PhD thesis, Stanford University, Palo Alto, CA, 1985. UMI Order Number: AAT 8506171.

[75] Clarkson, K. L. *Algorithms for Closest-Point Problems (Computational Geometry)*. PhD thesis, Stanford University, Stanford, CA, USA, 1985. AAT 8506171.

[76] Cleveland, W. S. Graphs in scientific publications. *The American Statistician 38*, 4 (1984), 261–269.

[77] Special issue: Digital libraries, Nov. 1996.

[78] Cohen, S., Nutt, W., and Sagic, Y. Deciding equivalances among conjunctive aggregate queries. *J. ACM 54*, 2 (Apr. 2007).

[79] Collell, G., Zhang, T., and Moens, M.-F. Imagined visual representations as multimodal embeddings. In *AAAI* (2017).

[80] Conti, M., Di Pietro, R., Mancini, L. V., and Mei, A. (new) distributed data source verification in wireless sensor networks. *Inf. Fusion 10*, 4 (Oct. 2009), 342–353.

[81] Conti, M., Di Pietro, R., Mancini, L. V., and Mei, A. (old) distributed data source verification in wireless sensor networks. *Inf. Fusion 10*, 4 (2009), 342–353.

[82] Corbyn, Z. To be the best, cite the best, 2010.

[83] Crone, S. F., and Finlay, S. Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting 28*, 1 (2012), 224–238.

[84] XBOW sensor motes specifications, 2008. http://www.xbow.com.

[85] Culler, D., Estrin, D., and Srivastava, M. Overview of sensor networks. *IEEE Comput. 37*, 8 (Special Issue on Sensor Networks) (2004), 41–49.

[86] Currie, G. *An ontology of art*. Springer, 1989.

[87] Dave, A. *Application of convolutional neural network models for personality prediction from social media images and citation prediction for academic papers*. University of California, San Diego, 2016.

[88] Dekel, O., Keshet, J., and Singer, Y. Large margin hierarchical classification. In *Proceedings of the twenty-first international conference on Machine learning* (2004), p. 27.

[89] Delacre, M., Lakens, D., and Leys, C. Why psychologists should by default use welch's t-test instead of student's t-test. *International Review of Social Psychology 30*, 1 (2017).

[90] Deng, C., Chen, Z., Liu, X., Gao, X., and Tao, D. Triplet-based deep hashing network for cross-modal retrieval. *IEEE Transactions on Image Processing 27*, 8 (2018), 3893–3903.

[91] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR* (2009), Ieee, pp. 248–255.

[92] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[93] Dietrich, J. The importance of being first: Position dependent citation rates on arxiv: astro-ph. *Publications of the Astronomical Society of the Pacific 120*, 864 (2008), 224.

[94] Dijkstra, E. Go to statement considered harmful. In *Classics in software engineering (incoll)*. Yourdon Press, Upper Saddle River, NJ, USA, 1979, pp. 27–33.

[95] Dimitrovski, I., Kocev, D., Loskovska, S., and Džeroski, S. Hierarchical annotation of medical images. *Pattern Recognition 44*, 10-11 (2011), 2436–2449.

[96] Dimitrovski, I., Kocev, D., Loskovska, S., and Džeroski, S. Hierarchical classification of diatom images using ensembles of predictive clustering trees. *Ecological Informatics 7*, 1 (2012), 19–29.

[97] Dizaji, K. G., Herandi, A., Deng, C., Cai, W., and Huang, H. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *ICCV* (2017).

[98] Dong, Y., Johnson, R. A., and Chawla, N. V. Will this paper increase your h-index?: Scientific impact prediction. In *WSDM* (2015), ACM, pp. 149–158.

[99] Dong, Y., Ma, H., Shen, Z., and Wang, K. A century of science: Globalization of scientific collaborations, citations, and innovations. In *KDD* (2017), ACM, pp. 1437–1446.

[100] Doolittle, W. F. Phylogenetic classification and the universal tree. *Science 284*, 5423 (1999), 2124–2128.

[101] Dorfer, M., Schlüter, J., Vall, A., Korzeniowski, F., and Widmer, G. End-to-end cross-modality retrieval with cca projections and pairwise ranking loss. *International Journal of Multimedia Information Retrieval 7*, 2 (2018), 117–128.

[102] Douglass, B. P., Harel, D., and Trakhtenbrot, M. B. Statecarts in use: structured analysis and object-orientation. In *Lectures on Embedded Systems*, G. Rozenberg and F. W. Vaandrager, Eds., vol. 1494 of *Lecture Notes in Computer Science*. Springer-Verlag, London, 1998, pp. 368–394.

[103] Dreyfus, S. E. An appraisal of some shortest-path algorithms. *Operations research 17*, 3 (1969), 395–412.

[104] Drucker, S., and Fernandez, R. A unifying framework for animated and interactive unit visualizations.

[105] Dunlop, D. D., and Basili, V. R. Generalizing specifications for uniformly implemented loops. *ACM Trans. Program. Lang. Syst. 7*, 1 (Jan. 1985), 137–158.

[106] Editor, I., Ed. *The title of book one*, 1st. ed., vol. 9 of *The name of the series one*. University of Chicago Press, Chicago, 2007.

[107] Editor, I., Ed. *The title of book two*, 2nd. ed. The name of the series two. University of Chicago Press, Chicago, 2008, ch. 100.

[108] Eichhorn, G. An overview of the astrophysics data system. *Experimental Astronomy 5*, 3-4 (1994), 205–220.

[109] Elzer, S., Carberry, S., and Zukerman, I. The automated understanding of simple bar charts. *Artificial Intelligence 175*, 2 (2011), 526–555.

[110] Elzer, S., Carberry, S., Zukerman, I., Chester, D., Green, N., and Demir, S. A probabilistic framework for recognizing intention in information graphics. In *IJCAI* (2005), vol. 19, LAWRENCE ERLBAUM ASSOCIATES LTD, p. 1042.

[111] English, L. Digital exhaust: The most valuable asset your organization owns, but isn't using. *Forbes*.

[112] Espinoza-Arias, P., Poveda-Villalón, M., García-Castro, R., and Corcho, O. Ontological representation of smart city data: From devices to cities. *Applied Sciences 9*, 1 (2019), 32.

[113] Fang, J., Mitra, P., Tang, Z., and Giles, C. L. Table header detection and classification. In *AAAI* (2012), pp. 599–605.

[114] Farrar, D. E., and Glauber, R. R. Multicollinearity in regression analysis: the problem revisited. *The Review of Economic and Statistics* (1967), 92–107.

[115] Fawcett, T. W., and Higginson, A. D. Heavy use of equations impedes communication among biologists. *PNAS 109*, 29 (2012), 11735–11739.

[116] Fear, S. *Publication quality tables in LaTeX*, April 2005. http://www.ctan.org/pkg/booktabs.

[117] Fei-Fei, L., and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *CVPR* (2005), vol. 2, IEEE, pp. 524–531.

[118] Feng, S., Fu, P., and Zheng, W. A hierarchical multi-label classification method based on neural networks for gene function prediction. *Biotechnology & Biotechnological Equipment 32*, 6 (2018), 1613–1621.

[119] for Artificial Intelligence, A. I. Scienceparse version 2, 2017.

[120] Fortunato, S., Bergstrom, C. T., Börner, K., Evans, J. A., Helbing, D., Milojević, S., Petersen, A. M., Radicchi, F., Sinatra, R., Uzzi, B., et al. Science of science. *Science 359*, 6379 (2018), eaao0185.

[121] Fowlkes, E. B., and Mallows, C. L. A method for comparing two hierarchical clusterings. *Journal of the American statistical association 78*, 383 (1983), 553–569.

[122] François, O. Arbitrariness of peer review: A bayesian analysis of the nips experiment. *arXiv preprint arXiv:1507.06411* (2015).

[123] Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association 32*, 200 (1937), 675–701.

[124] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. Devise: A deep visual-semantic embedding model. In *NIPS* (2013).

[125] Futrelle, R. P., Shao, M., Cieslik, C., and Grimes, A. E. Extraction, layout analysis and classification of diagrams in pdf documents. In *ICDAR* (2003), IEEE, pp. 1007–1013.

[126] Garfield, E. The history and meaning of the journal impact factor. *Jama 295*, 1 (2006), 90–93.

[127] Geiger, D., and Meek, C. Structured variational inference procedures and their realizations (as incol). In *Proceedings of Tenth International Workshop on Artificial Intelligence and Statistics,* The Barbados. The Society for Artificial Intelligence and Statistics, Jan. 2005.

[128] Geng, Y., Chen, J., Chen, Z., Pan, J. Z., Ye, Z., Yuan, Z., Jia, Y., and Chen, H. Ontozsl: Ontology-enhanced zero-shot learning. *Proceedings of the Web Conference 2021* (2021).

[129] Gerndt, M. *Automatic Parallelization for Distributed-Memory Multiprocessing Systems*. PhD thesis, University of Bonn, Bonn, Germany, Dec. 1989.

[130] Ghosal, T., Verma, R., Ekbal, A., and Bhattacharyya, P. DeepSentiPeer: Harnessing sentiment in review texts to recommend peer review decisions. In *ACL* (Florence, Italy, July 2019), Association for Computational Linguistics.

[131] Giunchiglia, E., and Lukasiewicz, T. Coherent hierarchical multi-label classification networks. *NeurIPS 2020 33* (2020).

[132] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems* (2014), pp. 2672–2680.

[133] Goossens, M., Rahtz, S. P., Moore, R., and Sutor, R. S. *The Latex Web Companion: Integrating TEX, HTML, and XML*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[134] Gopal, S., and Yang, Y. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), pp. 257–265.

[135] Grechkin, M., Poon, H., and Howe, B. Ezlearn: Exploiting organic supervision in large-scale data annotation. In *IJCAI* (2018).

[136] Gu, Y., Wang, C., Ma, J., Nemiroff, R. J., Kao, D. L., and Parra, D. Visualization and recommendation of large image collections toward effective sensemaking. *Information Visualization 16*, 1 (2017), 21–47.

[137] Guérin, J., and Boots, B. Improving image clustering with multiple pretrained cnn feature extractors. In *BMVC* (2018).

[138] Guérin, J., Gibaru, O., Thiery, S., and Nyiri, E. Cnn features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700* (2017).

[139] Gundy, M. V., Balzarotti, D., and Vigna, G. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies* (Berkley, CA, 2007), WOOT '07, USENIX Association.

[140] Gundy, M. V., Balzarotti, D., and Vigna, G. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies* (Berkley, CA, 2008), WOOT '08, USENIX Association, pp. 99–100.

[141] Gundy, M. V., Balzarotti, D., and Vigna, G. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies* (Berkley, CA, 2009), WOOT '09, USENIX Association, pp. 90–100.

[142] Guo, X., Liu, X., Zhu, E., and Yin, J. Deep clustering with convolutional autoencoders. In *ICONIP* (2017).

[143] Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. Canonical correlation analysis: An overview with application to learning methods. *Neural computation 16*, 12 (2004), 2639–2664.

[144] Harel, D. Logics of programs: Axiomatics and descriptive power. MIT Research Lab Technical Report TR-200, Massachusetts Institute of Technology, Cambridge, MA, 1978.

[145] Harel, D. *First-Order Dynamic Logic*, vol. 68 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, NY, 1979.

[146] Hart, V. I. Pi is (still) wrong, 2011.

[147] Hartl, M. The Tau Manifesto, 2010.

[148] CodeBlue: Sensor networks for medical care, 2008. http://www.eecs.harvard.edu/mdw/ proj/codeblue/.

[149] Haslam, N., Ban, L., Kaufmann, L., Loughnan, S., Peters, K., Whelan, J., and Wilson, S. What makes an article influential? predicting impact in social and personality psychology. *Scientometrics 76*, 1 (2008), 169–185.

[150] He, J., and Chen, C. Temporal representations of citations for understanding the changing roles of scientific publications. *Frontiers in Research Metrics and Analytics 3* (2018), 27.

[151] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR* (2016), pp. 770–778.

[152] Heaven, D. Ai peer reviewers unleashed to ease publishing grind, 2018.

[153] Heering, J., and Klint, P. Towards monolingual programming environments. *ACM Trans. Program. Lang. Syst. 7*, 2 (Apr. 1985), 183–213.

[154] Herlihy, M. A methodology for implementing highly concurrent data objects. *ACM Trans. Program. Lang. Syst. 15*, 5 (November 1993), 745–770.

[155] Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M., Ali, M., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409* (2017).

[156] Higham, N. *Handbook of writing for the mathematical sciences*. Society for Industrial Mathematics, 1998.

[157] Hinton, G. E., and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science 313*, 5786 (2006), 504–507.

[158] Hirsch, J. E. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences 102*, 46 (2005), 16569–16572.

[159] Hoare, C. A. R. Chapter ii: Notes on data structuring. In *Structured programming (incoll)*, O. J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, Eds. Academic Press Ltd., London, UK, UK, 1972, pp. 83–174.

[160] Hochman, N., and Manovich, L. Zooming into an instagram city: Reading the local through social media. *First Monday 18*, 7 (2013).

[161] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[162] Hohenecker, P., and Lukasiewicz, T. Deep learning for ontology reasoning.

[163] Hojat, M., Gonnella, J. S., and Caelleigh, A. S. Impartial judgment by the "gatekeepers" of science: fallibility and accountability in the peer review process. *Advances in Health Sciences Education 8*, 1 (2003), 75–96.

[164] Hollis, B. S. *Visual Basic 6: Design, Specification, and Objects with Other*, 1st ed. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[165] Hörmander, L. *The analysis of linear partial differential operators. III*, vol. 275 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, Germany, 1985. Pseudodifferential operators.

[166] Hörmander, L. *The analysis of linear partial differential operators. IV*, vol. 275 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, Germany, 1985. Fourier integral operators.

[167] Howe, B., Lee, P.-s., Grechkin, M., Yang, S. T., and West, J. D. Deep mapping of the visual literature. In *The Web Conference* (2017), International World Wide Web Conferences Steering Committee, pp. 1273–1277.

[168] Hu, M., and Chen, S. Doubly aligned incomplete multi-view clustering. In *IJCAI* (2018).

[169] Hu, Y., Gripon, V., and Pateux, S. Exploiting unsupervised inputs for accurate few-shot classification.

[170] Hua, X., Nikolov, M., Badugu, N., and Wang, L. Argument mining for understanding peer reviews. In *ACL* (Minneapolis, Minnesota, June 2019), Association for Computational Linguistics, pp. 2131–2137.

[171] Huang, A. Similarity measures for text document clustering. In *NZCSRSC* (2008), pp. 49–56.

[172] Huang, W., Chen, E., Liu, Q., Chen, Y., Huang, Z., Liu, Y., Zhao, Z., Zhang, D., and Wang, S. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *CIKM 2019* (2019), pp. 1051–1060.

[173] Huang, W., Wu, Z., Mitra, P., and Giles, C. L. Refseer: A citation recommendation system. In *JCDL* (2014), IEEE Press, pp. 371–374.

[174] Hullman, J., and Bach, B. Picturing science: Design patterns in graphical abstracts. *DIAGRAMS* (2018).

[175] Hutto, C. J., and Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM* (2014).

[176] Ieee tcsc executive committee. In *Proceedings of the IEEE International Conference on Web Services* (Washington, DC, USA, 2004), ICWS '04, IEEE Computer Society, pp. 21–22.

[177] Isaac, A., and Summers, E. Skos simple knowledge organization system. *Primer, World Wide Web Consortium (W3C) 7* (2009).

[178] J.D. West, I. Wesley-Smith, and C.T. Bergstrom. A recommendation system based on hierarchical clustering of an article-level citation network. *IEEE Transactions on Big Data 2*, 2 (June 2016), 113–123.

[179] Jeh, G., and Widom, J. Simrank: a measure of structural-context similarity. In *KDD* (2002), ACM, pp. 538–543.

[180] Jeong, C., Jang, S., Shin, H., Park, E., and Choi, S. A context-aware citation recommendation model with bert and graph convolutional networks. *arXiv preprint arXiv:1903.06464* (2019).

[181] Jiang, L., and Usbeck, R. Knowledge graph question answering datasets and their generalizability: Are they enough for future research? *SIGIR* (2022).

[182] Jin, C., Mao, W., Zhang, R., Zhang, Y., and Xue, X. Cross-modal image clustering via canonical correlation analysis. In *AAAI* (2015).

[183] Johnson, J. M., and Khoshgoftaar, T. M. Survey on deep learning with class imbalance. *Journal of Big Data 6*, 1 (2019), 27.

[184] Jolliffe, I. Principal component analysis. In *International encyclopedia of statistical science*. 2011, pp. 1094–1096.

[185] Jones, E., Oliphant, T., Peterson, P., et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed <today>].

[186] Jones, T., Huggett, S., and Kamalski, J. Finding a way through the scientific literature: indexes and measures. *World neurosurgery 76*, 1-2 (2011), 36–38.

[187] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. Bag of tricks for efficient text classification. In *EACL* (2017).

[188] Jurgens, D., Kumar, S., Hoover, R., McFarland, D., and Jurafsky, D. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics 6* (2018), 391–406.

[189] Jurisch, M., and Igler, B. Graph-convolution-based classification for ontology alignment change prediction. In *DL4KG@ ESWC* (2019), pp. 11–20.

[190] Kampffmeyer, M., Chen, Y., Liang, X., Wang, H., Zhang, Y., and Xing, E. P. Rethinking knowledge graph propagation for zero-shot learning. In *CVPR 2019* (2019), pp. 11487–11496.

[191] Kang, D., Ammar, W., Dalvi, B., van Zuylen, M., Kohlmeier, S., Hovy, E., and Schwartz, R. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *NAACL* (New Orleans, USA, June 2018).

[192] Karpathy, A., and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *CVPR* (2015).

[193] Kataria, S., Browuer, W., Mitra, P., and Giles, C. L. Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In *AAAI* (2008), vol. 8, pp. 1169–1174.

[194] Kazakova, O., Lee, P. L., Oh, B. M., Yang, T. S., West, J., and Howe, B. Viziometrics: Identifying central figures in scientific papers.

[195] Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., and Farhadi, A. A diagram is worth a dozen images. In *ECCV* (2016), Springer, pp. 235–251.

[196] King, G., and Zeng, L. Logistic regression in rare events data. *Political analysis 9*, 2 (2001), 137–163.

[197] Kipf, T. N., and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[198] Kiros, R., Salakhutdinov, R., and Zemel, R. Multimodal neural language models. In *ICML 2014* (2014), PMLR, pp. 595–603.

[199] Kiros, R., Salakhutdinov, R., and Zemel, R. S. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics* (2014).

[200] Kirschmer, M., and Voight, J. Algorithmic enumeration of ideal classes for quaternion orders. *SIAM J. Comput. 39*, 5 (Jan. 2010), 1714–1747.

[201] Klimt, B., and Yang, Y. The enron corpus: A new dataset for email classification research. In *ECML 2004* (2004), Springer, pp. 217–226.

[202] Knuth, D. E. *Seminumerical Algorithms*. Addison-Wesley, 1981.

[203] Knuth, D. E. *Seminumerical Algorithms*, 2nd ed., vol. 2 of *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 10 Jan. 1981.

[204] Knuth, D. E. *The TEXbook*. Addison-Wesley, Reading, MA., 1984.

[205] Knuth, D. E. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.)*. Addison Wesley Longman Publishing Co., Inc., 1997.

[206] Knuth, D. E. *The Art of Computer Programming*, 3rd ed., vol. 1 of *Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., 1998. (book).

[207] Kong, C., Lin, D., Bansal, M., Urtasun, R., and Fidler, S. What are you talking about? text-to-image coreference. In *CVPR* (2014).

[208] Kong, W.-C. *E-commerce and cultural values*. IGI Publishing, Hershey, PA, USA, 2001, name of chapter: The implementation of electronic commerce in SMEs in Singapore (Inbook-w-chap-w-type), pp. 51–74.

[209] Kong, W.-C. The implementation of electronic commerce in smes in singapore (as incoll). In *E-commerce and cultural values*. IGI Publishing, Hershey, PA, USA, 2001, pp. 51–74.

[210] Kong, W.-C. Chapter 9. In *E-commerce and cultural values (Incoll-w-text (chap 9) 'title')*, T. Thanasankit, Ed. IGI Publishing, Hershey, PA, USA, 2002, pp. 51–74.

[211] Kong, W.-C. The implementation of electronic commerce in smes in singapore (incoll). In *E-commerce and cultural values*, T. Thanasankit, Ed. IGI Publishing, Hershey, PA, USA, 2003, pp. 51–74.

[212] Kong, W.-C. *E-commerce and cultural values - (InBook-num-in-chap)*. IGI Publishing, Hershey, PA, USA, 2004, ch. 9, pp. 51–74.

[213] Kong, W.-C. *E-commerce and cultural values (Inbook-text-in-chap)*. IGI Publishing, Hershey, PA, USA, 2005, chapter: The implementation of electronic commerce in SMEs in Singapore, pp. 51–74.

[214] Kong, W.-C. *E-commerce and cultural values (Inbook-num chap)*. IGI Publishing, Hershey, PA, USA, 2006, chapter (in type field) 22, pp. 51–74.

[215] Korach, E., Rotem, D., and Santoro, N. Distributed algorithms for finding centers and medians in networks. *ACM Trans. Program. Lang. Syst. 6*, 3 (July 1984), 380–401.

[216] Kornerup, J. Mapping powerlists onto hypercubes. Master's thesis, The University of Texas at Austin, 1994. (In preparation).

[217] Kosiur, D. *Understanding Policy-Based Networking*, 2nd. ed. Wiley, New York, NY, 2001.

[218] Kriegeskorte, N. Open evaluation: a vision for entirely transparent post-publication peer review and rating for science. *Frontiers in computational neuroscience 6* (2012), 79.

[219] Krisnadhi, A., Hu, Y., Janowicz, K., Hitzler, P., Arko, R., Carbotte, S., Chandler, C., Cheatham, M., Fils, D., Finin, T., et al. The geolink modular oceanography ontology. In *ISWC 2015* (2015), Springer, pp. 301–309.

[220] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

[221] Küçüktunç, O., Saule, E., Kaya, K., and Çatalyürek, Ü. V. Direction awareness in citation recommendation.

[222] Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly 2*, 1-2 (1955), 83–97.

[223] Kulmanov, M., Khan, M. A., and Hoehndorf, R. Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics 34*, 4 (2018), 660–668.

[224] Kurtz, M. J., and Henneken, E. A. Measuring metrics-a 40-year longitudinal cross-validation of citations, downloads, and peer review in astrophysics. *Journal of the Association for Information Science and Technology 68*, 3 (2017), 695–708.

[225] Lafi, S., and Kaneene, J. An explanation of the use of principal-components analysis to detect and correct for multicollinearity. *Preventive Veterinary Medicine 13*, 4 (1992), 261–275.

[226] Lamport, L. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, MA., 1986.

[227] Larkin, J. H., and Simon, H. A. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science 11*, 1 (1987), 65–100.

[228] Larochelle, H., Erhan, D., and Bengio, Y. Zero-data learning of new tasks. In *AAAI 2008* (2008), vol. 1, p. 3.

[229] Larsen, P., and Von Ins, M. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics 84*, 3 (2010), 575–603.

[230] Lau, J. H., and Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368* (2016).

[231] Le, Q., and Mikolov, T. Distributed representations of sentences and documents. In *ICML* (2014).

[232] Le, Q. V. Building high-level features using large scale unsupervised learning. In *ICASSP* (2013).

[233] Lee, D. D., and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature 401*, 6755 (1999), 788.

[234] Lee, D. D., and Seung, H. S. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (2001), pp. 556–562.

[235] Lee, J. Transcript of question and answer session. In *History of programming languages I (incoll)*, R. L. Wexelblat, Ed. ACM, New York, NY, USA, 1981, pp. 68–71.

[236] Lee, N. Interview with bill kinder: January 13, 2005. *Comput. Entertain. 3*, 1 (Jan.-March 2005).

[237] Lee, P., West, J., and Howe, B. Viziometrix: A platform for analyzing the visual information in big scholarly data. In *The Web Conference Workshop on BigScholar* (2016).

[238] Lee, P., West, J., and Howe, B. Viziometrics: Analyzing visual patterns in the scientific literature. *IEEE Transactions on Big Data* (2017).

[239] Lee, P., West, J. D., and Howe, B. Viziometrics: Analyzing visual information in the scientific literature. *IEEE Transactions on Big Data* (2017).

[240] Lee, P.-s., Yang, S. T., West, J. D., and Howe, B. Phyloparser: A hybrid algorithm for extracting phylogenies from dendrograms. In *ICDAR* (2017).

[241] Lehmann, T. M., Schubert, H., Keysers, D., Kohnen, M., and Wein, B. B. The irma code for unique classification of medical images. In *Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation* (2003), vol. 5033, International Society for Optics and Photonics, pp. 440–451.

[242] Leonardi, P. M. Covid-19 and the new technologies of organizing: digital exhaust, digital footprints, and artificial intelligence in the wake of remote work. *Journal of Management Studies 58*, 1 (2021), 249.

[243] Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research 5*, Apr (2004), 361–397.

[244] Leydesdorff, L., and Zhou, P. Nanotechnology as a field of science: Its delineation in terms of journals and patents. *Scientometrics 70*, 3 (2007), 693–713.

[245] Li, C.-L., Buyuktur, A. G., Hutchful, D. K., Sant, N. B., and Nainwal, S. K. Portalis: using competitive online interactions to support aid initiatives for the homeless. In *CHI '08 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2008), ACM, pp. 3873–3878.

[246] Li, D., and Agha, L. Big names or big ideas: Do peer-review panels select the best science proposals? *Science 348*, 6233 (2015), 434–438.

[247] Li, Y., Wang, S., Umarov, R., Xie, B., Fan, M., Li, L., and Gao, X. Deepre: sequence-based enzyme ec number prediction by deep learning. *Bioinformatics 34*, 5 (2018), 760–769.

[248] Lin, J. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory 37*, 1 (1991), 145–151.

[249] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV* (2014).

[250] Lin, Y., Dong, X., Zheng, L., Yan, Y., and Yang, Y. A bottom-up clustering approach to unsupervised person re-identification. In *AAAI* (2019), vol. 2, pp. 1–8.

[251] Linhart, J. M. Algorithm 885: Computing the logarithm of the normal distribution. *ACM Transactions on Mathematical Software 35* (2008), Article 20.

[252] Linhart, J. M. Teaching writing and communication in a mathematical modeling course. *PRIMUS 24*, 7 (2014), 594–607.

[253] Liu, H., Kong, X., Bai, X., Wang, W., Bekele, T. M., and Xia, F. Context-based collaborative filtering for citation recommendation. *IEEE Access 3* (2015), 1695–1703.

[254] Liu, J., Wang, C., Gao, J., and Han, J. Multi-view clustering via joint nonnegative matrix factorization. In *SDM* (2013).

[255] Liu, X., Yu, Y., Guo, C., Sun, Y., and Gao, L. Full-text based context-rich heterogeneous network mining approach for citation recommendation. In *JCDL* (2014), IEEE Press, pp. 361–370.

[256] Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory 28*, 2 (1982), 129–137.

[257] López, F., and Strube, M. A fully hyperbolic neural model for hierarchical multi-class classification. In *EMNLP 2020* (2020), pp. 460–475.

[258] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2 (2004), 91–110.

[259] Lu, X., Wang, J., Mitra, P., and Giles, C. L. Automatic extraction of data from 2-d plots in documents. In *ICDAR* (2007), vol. 1, IEEE, pp. 188–192.

[260] Luo, C., Zhan, J., Xue, X., Wang, L., Ren, R., and Yang, Q. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *International Conference on Artificial Neural Networks* (2018), Springer, pp. 382–391.

[261] M. Rosvall, A.V. Esquivel, A. Lancichinetti, J.D. West, and R. Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature Communications 5*, 1 (2014).

[262] Maaten, L. Learning a parametric embedding by preserving local structure. In *AISTATS* (2009).

[263] Maaten, L. v. d., and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605.

[264] Madjarov, G., Vidulin, V., Dimitrovski, I., and Kocev, D. Web genre classification via hierarchical multi-label classification. In *IDEAL 2015* (2015), Springer, pp. 9–17.

[265] Madjarov, G., Vidulin, V., Dimitrovski, I., and Kocev, D. Web genre classification with methods for structured output prediction. *Information Sciences 503* (2019), 551–573.

[266] Mann, H. B., and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics* (1947), 50–60.

[267] Mansfield, E. R., and Helms, B. P. Detecting multicollinearity. *The American Statistician 36*, 3a (1982), 158–160.

[268] Mantel, N. The detection of disease clustering and a generalized regression approach. *Cancer research 27*, 2 Part 1 (1967), 209–220.

[269] Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., and Yuille, A. Deep captioning with multimodal recurrent neural networks (m-rnn). In *ICLR* (2015).

[270] Marshakova, I. Co-citation in scientific literature: A new measure of the relationship between publications.". *Scientific and Technical Information Serial of VINITI 6* (1973), 3–8.

[271] Masera, L., and Blanzieri, E. Awx: An integrated approach to hierarchical-multilabel classification. In *ECML-PKDD 2018* (2018), Springer, pp. 322–336.

[272] Masic, I. The importance of proper citation of references in biomedical articles. *Acta Informatica Medica 21*, 3 (2013), 148.

[273] McCracken, D. D., and Golden, D. G. *Simplified Structured COBOL with Microsoft/MicroFocus COBOL*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[274] McInnes, L., and Healy, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[275] McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A., and Riedl, J. On the recommending of citations for research papers. In *CSCW* (2002), ACM, pp. 116–125.

[276] Meding, K., Buschoff, L. M. S., Geirhos, R., and Wichmann, F. A. Imagenet suffers from dichotomous data difficulty. *NeurIPS 2021 Workshop on ImageNet PPF* (2021).

[277] Mewes, H.-W., Hani, J., Pfeiffer, F., and Frishman, D. Mips: a database for protein sequences and complete genomes. *Nucleic Acids Research 26*, 1 (1998), 33–37.

[278] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (2013), pp. 3111–3119.

[279] Miller, G. A. *WordNet: An electronic lexical database*. MIT press, 1998.

[280] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. Distant supervision for relation extraction without labeled data. In *ACL* (2009).

[281] Mons, B., and Velterop, J. Nano-publication in the e-science era. In *SWASD* (2009).

[282] Montieri, A., Ciuonzo, D., Bovenzi, G., Persico, V., and Pescapé, A. A dive into the dark web: Hierarchical traffic classification of anonymity tools. *IEEE Transactions on Network Science and Engineering 7*, 3 (2019), 1043–1054.

[283] Moons, E., Tuytelaars, T., and Moens, M.-F. Text-enriched representations for news image classification. *Companion Proceedings of the The Web Conference 2018* (2018).

[284] Mounce, R., Murray-Rust, P., and Wills, M. A machine-compiled microbial supertree from figure-mining thousands of papers. *Research Ideas and Outcomes 3* (2017), e13589.

[285] Muennighoff, N. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904* (2022).

[286] Mullender, S., Ed. *Distributed systems (2nd Ed.)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1993.

[287] Mumford, E. Managerial expert systems and organizational change: some critical research issues. In *Critical issues in information systems research (incoll)*. John Wiley & Sons, Inc., New York, NY, USA, 1987, pp. 135–155.

[288] Murthy, D. Towards a sociological understanding of social media: Theorizing twitter. *Sociology 46*, 6 (2012), 1059–1073.

[289] Nakano, F. K., Lietaert, M., and Vens, C. Machine learning for discovering missing or wrong protein function annotations. *BMC bioinformatics 20*, 1 (2019), 1–32.

[290] Natarajan, A., Motani, M., de Silva, B., Yap, K., and Chua, K. C. Investigating network architectures for body sensor networks. In *Network Architectures* (Dayton, OH, 2007), G. Whitcomb and P. Neece, Eds., Keleuven Press, pp. 322–328.

[291] Nathan Silberman, Derek Hoiem, P. K., and Fergus, R. Indoor segmentation and support inference from rgbd images. In *ECCV* (2012).

[292] Nelson, D. L., Reed, V. S., and Walling, J. R. Pictorial superiority effect. *Journal of Experimental Psychology: Human Learning and Memory 2*, 5 (1976), 523.

[293] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. Multimodal deep learning. In *ICML 2011* (2011).

[294] Nguyen, G., Worring, M., et al. Similarity based visualization of image collections.

[295] Nielson, F. Program transformations in a denotational setting. *ACM Trans. Program. Lang. Syst. 7*, 3 (July 1985), 359–379.

[296] Novak, D. Solder man. In *ACM SIGGRAPH 2003 Video Review on Animation theater Program: Part I - Vol. 145 (July 27–27, 2003)* (New York, NY, March 21, 2008 2003), ACM Press, p. 4.

[297] Nyberg, K., Raiko, T., Tiinanen, T., and Hyvönen, E. Document classification utilising ontologies and relations between documents. In *Proceedings of the eighth workshop on mining and learning with graphs* (2010), pp. 86–93.

[298] Nyberg, K., Raiko, T., Tiinanen, T., and Hyvönen, E. Document classification utilising ontologies and relations between documents. 86–93.

[299] Obama, B. A more perfect union. Video, Mar. 2008.

[300] Obeid, C., Lahoud, I., Khoury, H. E., and Champin, P.-A. Ontology-based recommender system in higher education. *Companion Proceedings of the The Web Conference 2018* (2018).

[301] Oberdörster, G., Oberdörster, E., and Oberdörster, J. Nanotoxicology: an emerging discipline evolving from studies of ultrafine particles. *Environmental health perspectives 113*, 7 (2005), 823.

[302] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics 20*, 17 (2004), 3045–3054.

[303] Ojala, T., Pietikainen, M., and Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence 24*, 7 (2002), 971–987.

[304] Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

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

[306] Pan, S. J., and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering 22*, 10 (2010), 1345–1359.

[307] Park, D. K., Jeon, Y. S., and Won, C. S. Efficient use of local edge histogram descriptor. In *Proceedings of the 2000 ACM workshops on Multimedia* (2000), pp. 51–54.

[308] Parolo, P. D. B., Pan, R. K., Ghosh, R., Huberman, B. A., Kaski, K., and Fortunato, S. Attention decay in science. *Journal of Informetrics 9*, 4 (2015), 734–745.

[309] Partalas, I., Kosmopoulos, A., Baskiotis, N., Artieres, T., Paliouras, G., Gaussier, E., Androutsopoulos, I., Amini, M.-R., and Galinari, P. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581* (2015).

[310] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[311] Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *EMNLP* (2014), pp. 1532–1543.

[312] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *NAACL* (2018).

[313] Petrie, C. J. New algorithms for dependency-directed backtracking (master's thesis). Tech. rep., Austin, TX, USA, 1986.

[314] Petrie, C. J. New algorithms for dependency-directed backtracking (master's thesis). Master's thesis, University of Texas at Austin, Austin, TX, USA, 1986.

[315] Poker-Edge.Com. Stats and analysis, Mar. 2006.

[316] Portenoy, J., and West, J. D. Using mathematical jargon to characterize differences between fields in the arxiv. In *International Journal on Digital Libraries*.

[317] Price, E. The nips experiment, 2014.

[318] Pulverer, B. Transparency showcases strength of peer review. *Nature 468*, 7320 (2010), 29.

[319] Qaraei, M., Schultheis, E., Gupta, P., and Babbar, R. Convex surrogates for unbiased loss functions in extreme classification with missing labels. *Proceedings of the Web Conference 2021* (2021).

[320] Ramos, J., et al. Using tf-idf to determine word relevance in document queries. In *iCML* (2003), vol. 242, pp. 133–142.

[321] Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. Collecting image annotations using amazon's mechanical turk. In *NAACL* (2010).

[322] Ray Choudhury, S., and Giles, C. L. An architecture for information extraction from figures in digital libraries. In *The Web Conference* (2015), ACM, pp. 667–672.

[323] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *CVPR* (2016), pp. 779–788.

[324] Reid, B. K. A high-level approach to computer document formatting. In *Proceedings of the 7th Annual Symposium on Principles of Programming Languages* (New York, Jan. 1980), ACM, pp. 24–31.

[325] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. *Recommender systems handbook*. Springer, 2015.

[326] Rohlf, F. J., and Fisher, D. R. Tests for hierarchical structure in random data sets. *Systematic Biology 17*, 4 (1968), 407–412.

[327] Rosengren, K. E. *Communication: an introduction*. Sage, 1999.

[328] Rous, B. The enabling of digital libraries. *Digital Libraries 12*, 3 (July 2008). To appear.

[329] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature 323*, 6088 (1986), 533.

[330] Ruxton, G. D. The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test. *Behavioral Ecology 17*, 4 (2006), 688–690.

[331] Saeedi, M., Zamani, M. S., and Sedighi, M. A library-based synthesis methodology for reversible logic. *Microelectron. J. 41*, 4 (Apr. 2010), 185–194.

[332] Saeedi, M., Zamani, M. S., Sedighi, M., and Sasanian, Z. Synthesis of reversible circuit using cycle-based approach. *J. Emerg. Technol. Comput. Syst. 6*, 4 (Dec. 2010).

[333] Sagi, T., Lehahn, Y., Bar, K., and Miller, L. A. Artificial intelligence for ocean science data integration: current state, gaps, and way forward. *Elementa: Science of the Anthropocene 8* (2020).

[334] Salas, S., and Hille, E. *Calculus: One and Several Variable*. John Wiley and Sons, New York, 1978.

[335] Santini, A. The importance of referencing. *Journal of critical care medicine (Universitatea de Medicina si Farmacie din Targu-Mures) 4*, 1 (2018), 3–4.

[336] Satterthwaite, F. E. An approximate distribution of estimates of variance components. *Biometrics bulletin 2*, 6 (1946), 110–114.

[337] Savva, M., Kong, N., Chhajta, A., Fei-Fei, L., Agrawala, M., and Heer, J. Revision: Automated classification, analysis and redesign of chart images. In *UIST* (2011), ACM, pp. 393–402.

[338] Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., and Džeroski, S. Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC bioinformatics 11*, 1 (2010), 1–14.

[339] Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *ESWC 2018* (2018), Springer, pp. 593–607.

[340] Scientist, J. The fountain of youth, Aug. 2009. Patent No. 12345, Filed July 1st., 2008, Issued Aug. 9th., 2009.

[341] Sethi, A., Sankaran, A., Panwar, N., Khare, S., and Mani, S. Dlpaper2code: Auto-generation of code from deep learning research papers. In *AAAI* (2018).

[342] Sethi, A., Sankaran, A., Panwar, N., Khare, S., and Mani, S. Dlpaper2code: Auto-generation of code from deep learning research papers. In *AAAI* (2018).

[343] Shannon, C. E., and Weaver, W. *The mathematical theory of communication*. University of Illinois press, 1998.

[344] Shao, M., and Futrelle, R. P. Recognition and classification of figures in pdf documents. In *GREC* (2005), Springer.

[345] Siegel, N., Horvitz, Z., Levin, R., Divvala, S., and Farhadi, A. Figureseer: Parsing result-figures in research papers. In *ECCV* (2016), Springer, pp. 664–680.

[346] Silla, C. N., and Freitas, A. A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery 22*, 1-2 (2011), 31–72.

[347] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[348] Small, H. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the Association for Information Science and Technology 24*, 4 (1973), 265–269.

[349] Smith, L. D., Best, L. A., Stubbs, D. A., Archibald, A. B., and Roberson-Nay, R. Constructing knowledge: The role of graphs and tables in hard and soft psychology. *American Psychologist 57*, 10 (2002), 749.

[350] Smith, S. W. An experiment in bibliographic mark-up: Parsing metadata for xml export. In *Proceedings of the 3rd. annual workshop on Librarians and Computers* (Milan Italy, 2010), R. N. Smythe and A. Noble, Eds., vol. 3 of *LAC '10*, Paparazzi Press, pp. 422–431.

[351] Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. Zero-shot learning through cross-modal transfer. In *NeurIPS 2013* (2013), pp. 935–943.

[352] Soergel, D., Saunders, A., and McCallum, A. Open scholarship and peer review: a time for experimentation. In *ICML 2013 Peer Review Workshop* (2013).

[353] Sokal, R. R. A statistical method for evaluating systematic relationship. *University of Kansas science bulletin 28* (1958), 1409–1438.

[354] Song, M., Lim, S., Kang, D., and Lee, S. Ontology-based automatic classification of web documents. In *ICIC 2006* (2006), Springer, pp. 690–700.

[355] Song, M.-H., Lim, S.-Y., Kang, D.-J., and Lee, S.-J. Automatic classification of web pages based on the concept of domain ontology. In *APSEC 2005* (2005), IEEE, pp. 7–pp.

[356] Spector, A. Z. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed., 2nd. ed. ACM Press, New York, NY, 1990, pp. 19–33.

[357] Steinbach, M., Ertöz, L., and Kumar, V. The challenges of clustering high dimensional data. In *New directions in statistical physics*. 2004, pp. 273–309.

[358] Stepišnik Perdih, T., Pollak, S., and Škrlj, B. Jsi at the finsim-2 task: Ontology-augmented financial concept classification. In *The Web Conference 2021* (2021), pp. 298–301.

[359] Strobelt, H., Oelke, D., Rohrdantz, C., Stoffel, A., Keim, D. A., and Deussen, O. Document cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1145–1152.

[360] Strohman, T., Croft, W. B., and Jensen, D. Recommending citations for academic papers. In *SIGIR* (2007), ACM, pp. 705–706.

[361] Student. The probable error of a mean. *Biometrika* (1908), 1–25.

[362] Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 843–852.

[363] Tahamtan, I., and Bornmann, L. Core elements in the process of citing publications: Conceptual overview of the literature. *Journal of Informetrics 12*, 1 (2018), 203–216.

[364] Tao, H., Hou, C., Liu, X., Yi, D., and Zhu, J. Reliable multi-view clustering. In *AAAI* (2018).

[365] Taylor, D. M. The appropriate use of references in a scientific research paper. *Emergency Medicine 14*, 2 (2002), 166–170.

[366] Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *NeuRIPS Dataset and Benchmark Track* (2021).

[367] Thornburg, H. Introduction to bayesian statistics, Mar. 2001.

[368] Tian, K., and Chen, H. aiai at the finsim-2 task: Finance domain terms automatic classification via word ontology and embedding. In *The Web Conference 2021* (2021), pp. 320–322.

[369] Tirilly, P., Claveau, V., and Gros, P. Language modeling for bag-of-visual words image categorization. In *CIVR* (2008), ACM, pp. 249–258.

[370] Tsai, Y.-H. H., Huang, L.-K., and Salakhutdinov, R. Learning robust visual-semantic embeddings. In *ICCV* (2017).

[371] Tseng, B.-H., Shen, S.-S., Lee, H.-Y., and Lee, L.-S. Towards machine comprehension of spoken content: Initial toefl listening comprehension test by machine. *Interspeech* (2016).

[372] Tsutsui, S., and Crandall, D. A data driven approach for compound figure separation using convolutional neural networks. *arXiv preprint arXiv:1703.05105* (2017).

[373] Tsutsui, S., and Crandall, D. J. A data driven approach for compound figure separation using convolutional neural networks. *CoRR abs/1703.05105* (2017).

[374] Institutional members of the TEX users group, 2017.

[375] Tzamaloukas, A., and Garcia-Luna-Aceves, J. J. Channel-hopping multiple access. Tech. Rep. I-CA2301, Department of Computer Science, University of California, Berkeley, CA, 2000.

[376] Ueki, K. Survey of visual-semantic embedding methods for zero-shot image retrieval. *ICMLA 2021* (2021).

[377] Van Noorden, R. Global scientific output doubles every nine years: News blog. nature news blog, 2014.

[378] Vens, C., Struyf, J., Schietgat, L., Džeroski, S., and Blockeel, H. Decision trees for hierarchical multi-label classification. *Machine learning 73*, 2 (2008), 185.

[379] Veytsman, B. acmart—Class for typesetting publications of ACM.

[380] Vilhena, D., Foster, J., Rosvall, M., West, J., Evans, J., and Bergstrom, C. Finding cultural holes: How structure and culture diverge in networks of scholarly communication. *Sociological Science 1* (2014), 221–238.

[381] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research 11*, Dec (2010), 3371–3408.

[382] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems 29* (2016).

[383] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset.

[384] Wang, C., Reese, J. P., Zhang, H., Tao, J., Gu, Y., Ma, J., and Nemiroff, R. J. Similarity-based visualization of large image collections. *Information Visualization 14*, 3 (2015), 183–203.

[385] Wang, H., Zong, L., Liu, B., Yang, Y., and Zhou, W. Spectral perturbation meets incomplete multi-view data. In *IJCAI* (2018).

[386] Wang, L., Li, Y., and Lazebnik, S. Learning deep structure-preserving image-text embeddings. In *CVPR* (2016).

[387] Wang, W., Arora, R., Livescu, K., and Bilmes, J. On deep multi-view representation learning. In *ICML* (2015).

[388] Wang, X., Ye, Y., and Gupta, A. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR 2018* (2018), pp. 6857–6866.

[389] Ware, C. *Information visualization: perception for design*. Elsevier, 2012.

[390] Wehrmann, J., Cerri, R., and Barros, R. Hierarchical multi-label classification networks. In *International Conference on Machine Learning* (2018), pp. 5075–5084.

[391] Welch, B. L. The generalization ofstudent's' problem when several different population variances are involved. *Biometrika 34*, 1/2 (1947), 28–35.

[392] Wenzel, E. M. Three-dimensional virtual acoustic displays. In *Multimedia interface design (incoll)*. ACM, New York, NY, USA, 1992, pp. 257–288.

[393] Werneck, R., Setubal, J. a., and da Conceicão, A. (new) finding minimum congestion spanning trees. *J. Exp. Algorithmics 5* (Dec. 2000).

[394] Werneck, R., Setubal, J. a., and da Conceicão, A. (old) finding minimum congestion spanning trees. *J. Exp. Algorithmics 5* (2000), 11.

[395] West, J., and Portenoy, J. Delineating fields using mathematical jargon. In *JCDL Workshop on BIRNDL* (2016).

[396] West, J. D., Wesley-Smith, I., and Bergstrom, C. T. A recommendation system based on hierarchical clustering of an article-level citation network. *IEEE Transactions on Big Data 2*, 2 (2016), 113–123.

[397] White, H. D., and Griffith, B. C. Author cocitation: A literature measure of intellectual structure. *Journal of the Association for Information Science and Technology 32*, 3 (1981), 163–171.

[398] White, K. E., Robbins, C., and Freyman, C. Science and engineering publication output trends: 2014 shows rise of developing country output while developed countries dominate highly cited publications.

[399] Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics 3* (2015), 345–358.

[400] Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. Towards universal paraphrastic sentence embeddings. In *ICLR* (2016).

[401] Wikipedia. Chaos Theory, 2012.

[402] Wikipedia. Fractal, 2012.

[403] Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[404] Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., and Schiele, B. Latent embeddings for zero-shot classification. In *CVPR 2016* (2016), pp. 69–77.

[405] Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep embedding for clustering analysis. In *ICML* (2016).

[406] Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. *NeuRIPS 2020 33* (2020), 6256–6268.

[407] Xu, C., and Geng, X. Hierarchical classification based on label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 5533–5540.

[408] Xu, C., Tao, D., and Xu, C. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing 24*, 12 (2015), 5812–5825.

[409] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *ICML* (2015), pp. 2048–2057.

[410] Xue, X., Jiang, C., Yang, C., Zhu, H., and Hu, C. Artificial neural network based sensor ontology matching technique. *Companion Proceedings of the Web Conference 2021* (2021).

[411] Yan, F., and Mikolajczyk, K. Deep correlation for matching images and text. In *CVPR* (2015).

[412] Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering.

[413] Yang, J., Parikh, D., and Batra, D. Joint unsupervised learning of deep representations and image clusters. In *CVPR* (2016).

[414] Yang, S., Huang, K.-H., and Howe, B. Multidec: Multi-modal clustering of image-caption pairs. *arXiv preprint arXiv:1901.01860* (2019).

[415] Yang, S., Huang, K.-H., and Howe, B. Multidec: Multi-modal clustering of image-caption pairs, 2019.

[416] Yang, S. T., and Howe, B. Surj: Ontological learning for fast, accurate, and robust hierarchical multi-label classification.

[417] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *NeuRIPS 2019 32* (2019).

[418] Yoon, J., and Chung, E. An investigation on graphical abstracts use in scholarly articles. *International Journal of Information Management*, 1 (2017), 1371–1379.

[419] Yu, T., Yu, G., Li, P.-Y., and Wang, L. Citation impact prediction for scientific papers using stepwise regression analysis. *Scientometrics 101*, 2 (2014), 1233–1252.

[420] Zhang, C., Fu, H., Liu, S., Liu, G., and Cao, X. Low-rank tensor constrained multiview subspace clustering. In *ICCV* (2015).

[421] Zhang, L., Xiang, T., and Gong, S. Learning a deep embedding model for zero-shot learning. In *CVPR 2017* (2017), pp. 2021–2030.

[422] Zhang, W., Deakin, J., Higham, N. J., and Wang, S. Etymo: A new discovery engine for ai research. WWW Demo.

[423] Zhang, Z., Liu, L., Shen, F., Shen, H. T., and Shao, L. Binary multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence* (2018).

[424] Zhao, H., Ding, Z., and Fu, Y. Multi-view clustering via deep matrix factorization. In *AAAI* (2017).

[425] Zhelezniak, V., Savkov, A., Shen, A., Moramarco, F., Flann, J., and Hammerla, N. Y. Don't settle for average, go for the max: Fuzzy sets and max-pooled word vectors. In *ICLR* (2019).

[426] Zhong, X., and Rajapakse, J. C. Graph embeddings on gene ontology annotations for protein–protein interaction prediction. *BMC bioinformatics 21*, 16 (2020), 1–17.

[427] Zhou, G., Lu, J., Wan, C.-Y., Yarvis, M. D., and Stankovic, J. A. *Body Sensor Networks*. MIT Press, Cambridge, MA, 2008.

[428] Zhou, G., Wu, Y., Yan, T., He, T., Huang, C., Stankovic, J. A., and Abdelzaher, T. F. A multifrequency mac specially designed for wireless sensor network applications. *ACM Trans. Embed. Comput. Syst. 9*, 4 (April 2010), 39:1–39:41.

[429] Zhu, X. P., and Ban, Z. Citation count prediction based on academic network features. In *AINA* (2018), IEEE, pp. 534–541.

[430] Zitt, M., and Bassecoulard, E. Delineating complex scientific fields by an hybrid lexical-citation method: An application to nanosciences. *Information processing & management 42*, 6 (2006), 1513–1531.

[431] Zou, Z., Tian, S., Gao, X., and Li, Y. mldeepre: Multi-functional enzyme function prediction with hierarchical multi-label deep learning. *Frontiers in genetics 9* (2019), 714.

[432] Zuckerkandl, E., and Pauling, L. Evolutionary divergence and convergence in proteins. In *Evolving genes and proteins*. Elsevier, 1965, pp. 97–166.