Augmenting Structure with Text for Improved Graph Learning

by

Tara L. Safavi

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2022

Doctoral Committee:

Professor Danai Koutra, Chair
Dr. Paul Bennett, Microsoft Research
Professor Kevyn Collins-Thompson
Professor Lu Wang

Tara L. Safavi
tsafavi@umich.edu
ORCID iD: 0000-0002-3553-4331

ACKNOWLEDGMENTS

First and foremost, I would like to thank my adviser, Danai Koutra, for her invaluable guidance over the past seven years. From starting my research journey as a undergraduate to defending my thesis in the midst of a pandemic, Danai has looked out for my professional and personal wellbeing every step of the way. She has inspired me greatly with her passion for research and technical brilliance, as well as her generosity and kindness toward her students. I feel confident that the lessons I have learned from Danai will guide me throughout my research career.

Next, I would like to thank my thesis committee members Paul Bennett, Kevyn Collins-Thompson, and Lu Wang for their excellent feedback and thought-provoking questions, which have greatly helped to improve this dissertation. I would like to extend a special thanks to Paul Bennett for being an incredible mentor over the course of two internships and beyond, always ready to help to foster new connections, bounce around ideas, or just chat about life. I still fondly remember how we managed to meet up and grab dinner in London, halfway around the world!

I am very privileged to have learned from world-class researchers throughout internships at Microsoft Research, Bloomberg, and the Allen Institute for Artificial Intelligence. In particular, I thank Adam Fourney, Robert Sim, Marcin Juraszek, Ned Friend, Edgar Meij, Jennifer Neville, Tobias Schnabel, Tom Hope, Doug Downey, and Hannaneh Hajishirzi for providing me amazing opportunities to grow as a researcher. I will always cherish my internships as some of the most unique, meaningful, and inspiring experiences in my academic career.

I am also very grateful to, and inspired by, my lab members and colleagues from the University of Michigan. Thank you in particular to GEMS Lab members Mark Heimann, Caleb Belth, Jing Zhu, and Jiong Zhu for the excellent discussions and fruitful collaborations. I learned a lot from you and can't wait to watch as your research careers flourish!

My PhD would not have been the same without the support of my amazing friends across Ann Arbor, London, Warsaw, and Seattle, as well as my parents and brother. Thank you for cheering me on no matter the distance between us, and for always reminding me of the important things in life. Finally, thank you to Michał Rybak for inspiring me every day to be the best person I can be. Kocham Cię!

FUNDING ACKNOWLEDGMENTS

This material is partially based upon work supported by the National Science Foundation under a Graduate Research Fellowship and CAREER Grant No. IIS 1845491, a Rackham Predoctoral Fellowship, Army Young Investigator Award No. W911NF1810397, the Advanced Machine Learning Collaborative Grant from Procter & Gamble, and Amazon, Google, and Facebook faculty awards. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or other funding parties.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Many important problems in machine learning and data mining, such as knowledge base reasoning, personalized entity recommendation, and scientific hypothesis generation, may be framed as learning and inference over a graph data structure. Such problems represent exciting opportunities for advancing graph learning, but also entail significant challenges. Because graphs are typically sparse and defined by a schema, they often do not fully capture the underlying complex relationships in the data. Models that combine graphs with rich auxiliary *textual* modalities have higher potential for expressiveness, but jointly processing such disparate modalities—that is, sparse structured relations and dense unstructured text—is not straightforward.

In this thesis, we consider the important problem of improving graph learning by combining structure and text. The first part of the thesis considers relational knowledge representation and reasoning tasks, demonstrating the great potential of pretrained contextual language models to add renewed depth and richness to graph-structured knowledge bases. The second part of the thesis goes beyond knowledge bases, toward improving graph learning tasks that arise in information retrieval and recommender systems by jointly modeling document interactions and content. Our proposed methodologies consistently improve accuracy over both single-modality and cross-modality baselines, suggesting that, with appropriately chosen inductive biases and careful model design, we can exploit the unique complementary aspects of structure and text to great effect.

CHAPTER 1

Introduction

Endowing machines with relational learning and reasoning skills over diverse inputs is a longstanding goal in artificial intelligence [Bush, 1945, Koller et al., 2007, Davis and Marcus, 2015, Lake et al., 2017, Battaglia et al., 2018, Hu et al., 2020]. Within this broad goal, the sub-discipline of *graph learning* focuses on relational prediction tasks over data that can be organized naturally into interconnected networks of nodes and edges. Example tasks commonly framed from a graph learning perspective include **(1)** knowledge base reasoning, or predicting novel factual relationships (edges) between pairs of entities or concepts (nodes) [Nickel et al., 2015]; **(2)** item recommendation, or inferring unseen affinity relationships (edges) among users and items (nodes) [Cooper et al., 2014]; and **(3)** drug repurposing, or connecting drugs and diseases (nodes) by potential treatment relationships (edges) [Nadkarni et al., 2021].

Learning over graphs is challenging for several reasons. In many cases, the underlying interactions between pairs of nodes cannot be fully observed or recorded. For example, relational knowledge bases (KBs) omit many basic facts about notable entities because KB curation is costly [Galárraga et al., 2017, Weikum et al., 2021]. In recommender systems, observed interactions between users and items—for example clicks, purchases, and ratings—are often sparse, since most users do not express their full preferences through online behavior [Wang et al., 2019c]. In drug repurposing, novel drug treatments must be reviewed and tested extensively before being recorded in biomedical databases, a lengthy and expensive process [Bonner et al., 2021].

More fundamentally, graphs are data representations intended to model a small, specific set of interaction phenomena according to a research question or prediction task of interest [Brugere et al., 2018]. Therefore, the types of meaning conveyed by a graph's nodes and edges are typically limited by a schema, which may not fully express the underlying semantics or complex relationships in the data [Davis et al., 1993]. It thus becomes natural to consider incorporating external features into graph datasets. In particular, we observe that many graphs of interest are partially derived from the Web and can be linked to online textual sources, which are often available freely and in abundance. For example, encyclopedic knowledge bases like Freebase [Bollacker et al., 2008] and Wikidata [Vrandečić and Krötzsch, 2014] link a majority of their entities to Wikipedia

Table 1.1: Thesis organization. We categorize **methodology contributions** as joint modeling (i.e., one model that takes both structure and text as input) and model fusion (i.e., multiple models trained over distinct structural and textual views of the data, and combined via ensembling). We categorize **resource contributions** as new datasets and new taxonomies. We categorize **evaluation tasks** as link prediction (LP) and node classification (NC) in a graph. For Part I, KRR refers to knowledge representation and reasoning.

Part	Contribution type		Task		Technical objective	Chapter
	Method	Resource	LP	NC		
I	-	Taxonomy	✓		Organize structure + text KRR strategies	3
	Joint model	-	✓		Infer negative commonsense knowledge	4
	Model fusion	Dataset	✓		Infer novel factual knowledge	5
II	Joint model	-	✓		Discover activities in personal information collections	6
	Model fusion	-		✓	Classify documents with cross-modal inputs	7

pages. Products in e-commerce interaction graphs are often associated with descriptions and reviews [Wan and McAuley, 2018]. Drugs and diseases in drug repurposing graphs can be linked to textual descriptions from online biomedical databases [Nadkarni et al., 2021].

Motivated by these observations, in this thesis we argue that **incorporating auxiliary text can address the challenge of sparsity in graphs and improve expressiveness in graph learning**. Whereas graphs provide a valuable "high-level" relational view of data, text naturally *complements* graphs by providing rich, lower-level unstructured context [Huang et al., 2020]. Of course, because these two data modalities are highly disparate [Halevy et al., 2003], we face a natural challenge of cross-modal integration. We thus contribute a suite of tools and techniques for text-augmented graph learning, including (1) new machine learning methodologies for joint learning, in which we optimize a single model that takes both relational structure and textual content as input; (2) new machine learning methodologies for model fusion, in which we optimize multiple machine learning models over various structural and textual "views" of the data, and develop novel approaches to combine these models' outputs; and (3) new resources such as taxonomies and datasets, toward accelerating research in text-augmented graph learning.

1.1 Organization

This thesis is organized into two parts. In the first part, we consider structure and text learning toward the goal of machine knowledge representation and reasoning, focusing on using deep contextual language models to augment relational world knowledge bases. In the second part, we go beyond knowledge bases, toward improving diverse information retrieval and recommendation systems by modeling document interactions and contents. Table 1.1 provides a high-level organization of the thesis.

1.1.1 Relational Knowledge Representation with Language Models

In this first part of the thesis, we consider fundamental problems in machine knowledge representation and reasoning. We begin by setting the stage in Chapter 3 with the following question:

> *What are state-of-the-art knowledge representation strategies and reasoning tasks that combine structure and text?*

To answer this question, we turn to pretrained contextual language models (LMs) based on the Transformer [Vaswani et al., 2017] neural network architecture. Contextual language models like BERT [Devlin et al., 2019] and GPT-3 [Brown et al., 2020], which model the conditional probability distributions of tokens in large text corpora derived from the Web, have shown to be capable of internalizing and expressing a degree of relational "knowledge" about the world, as surfaced through probing studies [Petroni et al., 2019] and downstream entity- and relation-centric evaluation tasks like question answering [Roberts et al., 2020]. We review these capabilities from a knowledge base (KB) perspective, proposing a taxonomy that organizes knowledge representation strategies in LMs by the level of KB supervision provided to the LM, from no KB supervision at all to entity- and relation-level supervision. For each level of supervision, we showcase exemplary methodologies and findings. We also discuss how graph learning methodologies may be informed by advancements in language modeling, leading the way into the next two chapters.

Chapter 4 addresses the following question:

> *How can we improve machine learning models' ability to discriminatively reason over everyday "commonsense" human knowledge?*

Our answer to this question builds on recent research in machine knowledge acquisition, which trains language models to automatically generate semi-structured statements in commonsense KBs, and ultimately uses these statements as training data for downstream reasoning tasks that require background world knowledge [Bosselut et al., 2019, Hwang et al., 2021]. While this line of research has proved highly successful, it has so far focused only on generating positive (true) KB statements, even though *negative* (false) statements are equally important for training discriminative models of world knowledge [Xiong et al., 2020a]. As a first step toward the latter, we propose NegatER, a joint learning framework that generates negative examples in commonsense KBs by mining the implicit knowledge stored in a language model's parameters. Experiments demonstrate that training models on NegatER-generated negative examples leads to statistically significant accuracy improvements in a challenging KB link prediction task compared to competitive data augmentation baselines. A human evaluation task also confirms that NegatER negatives are more grammatical and coherent than those generated by baselines.

Finally, Chapter 5 addresses the following question:

How can we efficiently and effectively combine structure and text to augment encyclopedic knowledge bases, which are typically highly incomplete?

We make multiple contributions in answering this question. Our first contribution is CoDEx, a new suite of cross-modal evaluation benchmarks for link prediction in encyclopedic KBs, also known as KB completion or KBC. To motivate the need for CoDEx, we review existing benchmarks and tasks in KB completion, concluding that no benchmark is suitable for the unique challenges of KBC across structure and text. To fill this gap, we propose CoDEx, a collection of three structural KBs linked to entity and relation aliases, descriptions, and Wikipedia page extracts. We analyze the structural and textual components of CoDEx extensively, and demonstrate the unique merits of CoDEx in terms of scope and difficulty compared to a widely-used KBC benchmark.

Having thoroughly established the value of CoDEx, we next propose CascadER, a new model fusion approach that combines structure-only KB embeddings with language models for KBC. CascadER is motivated by the observation that these two types of approach are complementary in terms of efficiency and effectiveness: KB embeddings are fast but do not benefit from all of the rich contextual information in text, whereas pretrained language models capture this nuance at the expense of efficiency [Wang et al., 2021a]. Building on this observation, CascadER orders the two types of models by efficiency such that language models are used to rerank small sets of outputs from KB embeddings—thus exploiting the expressiveness of LMs for KBC on promising subsets of the problem space, while avoiding their inefficiencies. We show that CascadER achieves consistent and appreciable gains over structure-only, text-only, *and* cross-modal baselines on multiple link prediction benchmarks, including but not limited to CoDEx. We also analyze CascadER empirically to demonstrate how it trades off effectiveness and efficiency, showing that it outperforms or matches a highly competitive ensembling baseline while improving efficiency by one or more orders of magnitude. Finally, we confirm a major hypotheses of this thesis, which is that combining structure and text leads to larger performance improvements on sparser graphs.

1.1.2 Document Interaction and Content Mining

In the second part of the thesis, we go beyond knowledge bases, considering diverse information retrieval and recommendation settings in which the interplay between structural interactions and textual content is crucial. In Chapter 6, we consider a unique setting in which we observe the digital trails and traces of individuals as both interaction (e.g., emails between people, clicks on files) and content (e.g., email subjects and bodies, file contents). We address the following question:

Can we model activities (projects, hobbies, tasks) in individuals' heterogeneous personal information collections (files, emails, contacts, etc) using both interaction structure and document contents to power personal search and recommendation?

To answer this, we propose the Personal Web, an interconnected view of personal information collections that integrates user-document interactions, inter-document interactions, and document contents that are relevant to people's activities (e.g., names of projects). We derive fast linear methods to learn and incrementally update entity representations in this graph view. We demonstrate the strengths of Personal Webs in two recommendation tasks framed as link prediction over the Personal Web. First, we formulate a unique personalized entity recommendation task in which we gather direct judgments from individuals over their own data. Second, we devise a larger-scale email recipient recommendation task using a public dataset. In both tasks, Personal Webs outperform diverse baselines according to standard ranking metrics, suggesting that accurately modeling the interplay of interaction and content is key to search and recommendation over personal information collections.

Finally, Chapter 7 addresses the following question:

> *How can we improve document classification accuracy by leveraging inter-document relationships and document contents?*

This question is common in information retrieval settings in which the document corpus has a natural underlying relational structure: For example, textual scientific articles indexed by scholarly search engines are linked by citations, and products indexed by e-commerce platforms are linked by co-purchase relationships. To solve this problem, we propose Late-Stage Fusion (LSF), an effective model fusion approach that combines the class predictions of graph learning models with non-relational, text-only models via a weighted majority vote. We provide theoretical and empirical analysis to show that LSF successfully models local class distributions patterns in a graph, surpassing the performance of any single-modality group of models. In extensive experiments, we confirm the superior and robust performance of LSF, which outperforms or matches 13 competitive baselines across document interaction graphs with varying degrees of class correlation.

1.2 Contributions

This thesis makes the following contributions:

- **Call to action**: This thesis invites the graph learning community to renew interest in combining structure and text for improved performance in fundamental graph learning tasks.
- **Bridging the gap**: This thesis introduces concepts from natural language processing and information retrieval into graph learning research, an important step toward bringing these sub-disciplines closer together and accelerating innovation and discovery.

- **Methodological advancements**: We develop new cross-modal graph learning methodologies that incorporate textual content to improve accuracy in the canonical graph learning tasks of link prediction and node classification. Our proposed approaches belong to two classes of methodology. The first is joint modeling, in which we optimize a single model that takes both structural and textual data as input. The second is model fusion, in which we optimize separate models over structural and textual views of the data, and devise methods to combine their outputs.

- **Evaluation insights**: We emphasize well-designed and comprehensive experimentation using both quantitative (task-based) and qualitative (human judgment-based) evaluation. In all of our experiments, our proposed approaches achieve consistent and appreciable gains over structure-only *and* text-only approaches, demonstrating the unique potential of these two modalities to complement each other.

- **New resources**: Toward increasing interest and participation in research at the intersection of graph and text learning, we introduce two new resources: A comprehensive taxonomy for structured knowledge representation in contextual language models, and a well-annotated knowledge base completion dataset spanning structure and text.

1.2.1 Research Impact

The research presented in this thesis has made the following impacts:

- **New research directions**: In recent years, several research groups across academia and industry have expressed interest in developing "personal knowledge graphs" to improve users' personalized digital experiences. The concept of the Personal Web, which we introduce in Chapter 6, has helped shape this emerging direction. We have presented this research as a keynote at workshops co-located with the Automatic Knowledge Base Construction (AKBC) conference and the Knowledge Graphs Conference (KGC), indicating that our work is actively influencing this research area.[1,2]

- **Industry applications**: The research directions discussed in Chapters 5, 6, and 7 were developed with industry collaborators. The work in Chapter 5 has particular impact in the context of biomedicine, as we show that our proposed approach CascadER can significantly improve performance in automated drug repurposing over knowledge bases, toward reducing the cost and streamlining the pipeline of drug repurposing [Bonner et al., 2021]. Chapters 6 and

[1] https://pkgs.ws/
[2] https://phkg.github.io/

7 have direct applications in commercial information retrieval and recommendation: Both were motivated by the need for efficient, large-scale graph learning approaches that incorporate textual content for business-critical goals like query understanding [Larson et al., 2020] and user modeling [Yang et al., 2021]. The Personal Webs work in Chapter 6 has also been covered in a highly-disseminated blog post on the Microsoft Research Blog.[3]

- **Widely-used resources**: While only recently released, our CoDEx benchmark proposed in Chapter 5 has been integrated into two major open-source libraries for knowledge representation: LibKGE [Broscheit, 2019] and PyKEEN [Ali et al., 2021]. Between these two libraries, CoDEx has over 1000 stars on GitHub, indicating a high level of interest from the community. Moreover, both our CoDEx benchmark and our survey presented in Chapter 3 were covered in widely-read "Knowledge Graphs for NLP" newsletter digests, increasing their visibility and reach across the NLP community.[4,5]

[3] https://tinyurl.com/yckwdujk
[4] https://migalkin.github.io/posts/2020/11/19/post/
[5] https://migalkin.github.io/posts/2021/11/14/post/

CHAPTER 2

Preliminaries

In this chapter, we review necessary preliminaries on machine learning over graphs and text.

2.1 Preliminaries on Graphs

2.1.1 Definitions and Notation

A **graph** or network is a data structure consisting of **nodes** or vertices and pairwise **edges** or links between nodes. A **singly-relational** graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph in which edges express only one semantic relationship type between nodes, consisting of a set of nodes $\mathcal{V}$ and a set of edges $(u, v) \in \mathcal{E} \in \mathcal{V} \times \mathcal{V}$. A **multi-relational** graph $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$ is a graph in which edges express multiple semantic relationship types, consisting of a set of nodes $\mathcal{V}$, a set of relation types or **relations** $\mathcal{R}$, and a set of ordered **triples** $(u, r, v) \in \mathcal{E} \in \mathcal{V} \times \mathcal{R} \times V$ expressing typed edges between pairs of nodes. Notice that a singly-relational graph is a special case of a multi-relational graph in which $|\mathcal{R}| = 1$ and all triples (u, r, v) contain the same relation type r. Figure 2.1 provides an example of a multi-relational graph expressing linguistic relationships between words and parts of speech.

Both singly- and multi-relational graphs may be undirected or directed, as well as unweighted or weighted. In an **undirected** graph, the presence of an edge between nodes u and v implies that the reverse edge of the same type between v and u also exists. By contrast, in a **directed** graph, the presence of an edge between u and v does not imply the presence of the reverse edge between v and u. In an **unweighted** graph, each edge is binary, expressing the presence or absence of a relationship between nodes u and v. In a **weighted** graph, each edge is associated with a real-valued weight $w_{u,v}$ expressing the strength of the relationship between nodes u and v.

Beyond set notation, a graph's edges may also be expressed with matrix notation. A singly-relational unweighted graph's edges may be encoded in a binary **adjacency matrix** $\mathbf{A} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ in which $A_{uv} = 1$ if there is an edge between nodes u and v, and 0 otherwise. Likewise, a multi-relational unweighted graph's edges may be encoded in a binary **adjacency tensor**

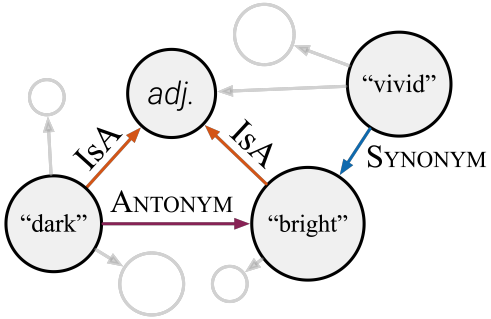


Figure 2.1: An example of a multi-relational graph expressing linguistic relationships between words and parts of speech. Nodes are the circles in the figure, edges are arrows, and relation types are the labels associated with the edges.

$\mathbf{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{R}|\times|\mathcal{V}|}$ in which $A_{urv} = 1$ if there is an edge of relation type r between nodes u and v, and 0 otherwise. If the graph is weighted, then its adjacency matrix or tensor contains real-valued rather than binary entries.

2.1.2 Linking Graphs to Text

In machine learning, graphs are used to represent complex real-world phenomena like social interactions, user-item affinities, and drug-disease treatments. Therefore, the nodes in graphs are typically linked to side **attributes**, which may be real-valued or categorical features, or raw data types like text or images. For example, given a graph expressing professional email interactions between people, "*email*" nodes in the graph may be associated with attributes like sent/received timestamps, subject lines, and bodies, and "*person*" nodes in the graph may be associated with profile attributes like name, location, and job title [Jin et al., 2019a].

As the focus of this thesis is text-augmented graph learning, we focus on graphs whose nodes are linked to textual attributes, for example email bodies in email interaction networks and product descriptions in user-item affinity graphs. In some cases (e.g., Chapters 6 and 7), we will preprocess the textual content associated with each node by extracting a d-dimensional textual feature vector for that node; we defer discussion of text featurization methods to Chapter 2.5. In these cases, we use the notation $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times d}$ to represent a **node feature matrix** in which row $\mathbf{x}_u$ maps node u to its textual feature vector. In other cases (e.g., Chapters 4 and 5), we will learn a direct mapping between each node's raw textual content and the output classification or regression value of interest, without preprocessing the data to extract feature vectors first. In these cases, we use the notation $[X_1, \ldots, X_{N_u}]$ to refer to a sequence of N_u words associated with node u.

2.2 Graph Learning Tasks

In this thesis, we consider two fundamental graph learning tasks: Link prediction and node classification.

2.2.1 Link Prediction

The link prediction task aims to augment a graph with new edges. Examples of tasks framed as link prediction in this thesis include predicting user-item affinities on digital platforms (e.g., recommendation), predicting novel drug treatments for diseases (e.g., hypothesis generation), and predicting recipients to emails (e.g., recipient suggestion).

In this thesis, we consider both ranking-based link prediction and classification-based link prediction. **Ranking-based link prediction** trains a machine learning model to score plausible unseen edges in a graph higher than implausible ones [Liben-Nowell and Kleinberg, 2007]. Evaluation is conducted as follows: Given a **query** node $u \in \mathcal{V}$ and optional query relation $r \in \mathcal{R}$ in the case of a multi-relational graph, score all potential **answer** nodes $v \in \mathcal{V}$ by the likelihood that they "answer the query"—that is, form a true edge in the graph. Performance is measured with mean reciprocal rank (**MRR**), the average reciprocal of each ground-truth answer entity's rank over all link prediction queries, and **hits@**k, the proportion of test queries for which the ground-truth answer entity is ranked in the top-k predicted answers.

While link prediction is most often framed as a ranking task, it has also been studied from a classification perspective [Socher et al., 2013, Li et al., 2016]. In **classification-based link prediction**, a machine learning model is trained to distinguish true unseen edges in a graph from false ones. Evaluation is conducted as follows: Given an unseen edge (u, v) in a singly-relational graph or (u, r, v) in a multi-relational graph, map the input edge to a binary label $\{0, 1\}$ that indicates whether the edge exists in the graph.

Link prediction may be studied under the transductive and inductive settings. In **transductive** link prediction, all entities and relations presented to a model at test time have been observed by the model during training. In **inductive** link prediction, the model may be presented entities and/or relations at test time that have not been observed during training. In this thesis, we will consider the transductive link prediction task in Chapters 5 and 6, and inductive link prediction in Chapter 4.

2.2.2 Node Classification

The second graph learning task considered in this thesis is node classification, which is formulated over the nodes rather than edges of the graph. The **node classification** task aims to classify nodes according to a predefined set of classes $\mathcal{C}$. The standard evaluation metric for node classification is

accuracy. Examples of node classification include categorizing documents in hyperlink graphs by topic (e.g., for search engine indexing), categorizing products in e-commerce networks by product type (e.g., for recommendation), and classifying emails in email interaction networks as spam.

Similar to link prediction, node classification has been studied in both transductive and inductive settings. The **transductive node classification** task is formulated as follows: Given a training set consisting of nodes $\mathcal{V}_{\text{train}} \subset \mathcal{V}$ and their labels, node features $\mathbf{X}$ for *all* $\mathcal{V}$ nodes, and the full $\mathcal{V} \times \mathcal{V}$ adjacency matrix $\mathbf{A}$, predict the labels of the remaining nodes $\mathcal{V}_{\text{test}} = \mathcal{V} \setminus \mathcal{V}_{\text{train}}$. By contrast, the **inductive node classification** task is formulated as follows: Given a training set consisting of nodes $\mathcal{V}_{\text{train}} \subset \mathcal{V}$ and their labels, node features $\mathbf{X}$ for the training nodes $\mathcal{V}_{\text{train}}$ only, and the adjacency matrix $\mathbf{A}_{\text{train}}$ across the training nodes $\mathcal{V}_{\text{train}}$ only, predict the labels of the test nodes $\mathcal{V}_{\text{test}} = \mathcal{V} \setminus \mathcal{V}_{\text{train}}$ using their features and structural connections, which were not observed at training time. In this thesis, we consider transductive node classification in Chapter 7, and leave inductive node classification for future work.

2.3 Graphs as Knowledge Representations

The first part of this thesis focuses on using attributed multi-relational graphs as machine knowledge representations. Formally, a **knowledge representation** is an internal mechanism or model within an intelligent agent that expresses what the agent knows about the external world. Knowledge representations are fundamentally tied to learning and reasoning [Davis et al., 1993], as they constrain how the agent can and cannot view the world. A knowledge representation is **relational** if it is capable of expressing relationships between the objects captured by the representation. Such relationships may have a variety of semantics, for example temporal, spatial, social, or causal.

In this thesis, we consider knowledge representations from the perspective of relational **knowledge bases** (KBs) or equivalently **knowledge graphs** (KGs). A KB is a graph in which nodes represent notable real-world **entities** or concepts (e.g., people, places, things), and edge types represents factual semantic relationships between entity pairs (e.g., social relationships, geographical relationships, taxonomic relationships). In this thesis, we will use the **(head, relation, tail)** ordered triple notation $(h, r, t) \in \mathcal{V} \times \mathcal{R} \times V$ to denote edges in KBs.

We consider two types of KBs under the umbrella of relational world knowledge. **Encyclopedic** or factual KBs store facts about typed, disambiguated entities. A well-known encyclopedic KB powering search and discovery on the Web is the public collaborative Wikidata knowledge base [Vrandečić and Krötzsch, 2014], which is a sister project to Wikipedia. By contrast, in **commonsense** KBs, "entities" are typically not disambiguated, and are represented by free-text phrases referring to, e.g., everyday objects, actions, and concepts. The lack of entity disambiguation means that commonsense KBs are technically semi-structured rather than fully-structured knowledge rep-

resentations. Such KBs may also be called "non-canonicalized" or "open" KBs [Broscheit et al., 2020a]. Examples of commonsense KBs include the crowdsourced ConceptNet [Liu and Singh, 2004, Speer et al., 2017] and ATOMIC [Sap et al., 2019, Hwang et al., 2021] KBs.

Note that knowledge bases are a special class of **heterogeneous information network** [Sun and Han, 2012, Shi et al., 2016], the latter of which is defined as a graph consisting of multiple node and edge types. However, whereas heterogeneous networks are generalized and may express any complex phenomenon (e.g., interacting agents on an e-commerce platform, treatment relationships between drugs and diseases), knowledge bases exclusively store factual knowledge about the world. Moreover, heterogeneous networks typically consist of relatively few node and edge types, on the order of ten or fewer, whereas knowledge bases may contain hundreds or thousands of entity and relation types. In practice, this means that different methods are required for machine learning over knowledge bases compared to established heterogeneous network mining approaches: The latter typically rely on handcrafted "meta-paths" of node and edge types to capture the graph's various semantic patterns [Sun and Han, 2012], an approach that is infeasible with a large number of types.

2.3.1 Knowledge Base Construction and Completion

Constructing a KB requires first defining a **schema** that specifies the entity types, relation types, and attribute fields captured by the representation, and the logical connections between these types. This step is usually led by domain experts [Suchanek et al., 2007, Vrandečić and Krötzsch, 2014, Ammar et al., 2018, Weikum et al., 2021]. The KB is then populated using manual and/or automated techniques. There are two fundamental goals in KB construction, which are in constant tension [Weikum et al., 2021]: **Precision** (correctness) and **recall** (coverage). The most popular KBs (Wikidata, ConceptNet) are curated primarily via humans and are therefore relatively high-precision. However, they tend to be low-recall for all but the most popular of entities, albeit continually expanding [Razniewski and Das, 2020].

Due to this incompleteness, there is great interest in automatically constructing and completing KBs. Automatic KB construction is a broad goal consisting of multiple sub-tasks [Weikum et al., 2021], many of which belong to the domain of information extraction from text (e.g., entity recognition, entity linking, relation extraction) [Ji and Grishman, 2011], which is out of scope of this thesis. However, there is also increasing interest from the graph learning community in developing models that learn over the KB's relational structure and attribute content to infer missing links between entities in the KB [Nickel et al., 2015]. From this perspective, the task of graph-based link prediction in knowledge bases is often referred to as **knowledge base completion**, or **KBC** [Bordes et al., 2013, Nickel et al., 2015, Ruffinelli et al., 2020].

Knowledge base completion is most often framed as a ranking task in the graph learning literature. Ranking-based KBC consists of two settings. In the first, given a tail query $(h, r, ?)$, score all entities $\hat{t} \in \mathcal{V}$ by their likelihood that they answer the query such that the gold tail entity t is ranked as high as possible. In the second, given a head query $(?, r, t)$, score all entities $\hat{h} \in \mathcal{V}$ by the likelihood that they answer the query. That said, KBC may also be framed as a classification task, as we do in Chapter 4. Under this framing, we classify novel triples (h, r, t) as true or false according to the knowledge statements that they convey.

2.4 Graph Representation Learning

The choice of data representation is a fundamental challenge in machine learning, as the data representation heavily influences how well the model can learn a mapping from input to output. Therefore, a long-standing question in machine learning is how to best capture the features of a dataset, most often in vector/matrix form, as most powerful data mining and processing algorithms expect vectorized data [Wasserman, 2010]. Whereas the early paradigm in machine learning was to manually select and extract a set of potentially salient features from the data, a common approach today is to *learn* the features of the data without manual feature engineering. Toward this goal, **representation learning** refers to iteratively optimizing the weights in one or more parameter vectors in order to minimize an objective or cost function that involves those parameters [Goodfellow et al., 2016]. After optimization is complete, the parameters are called **embeddings**, **latent representations**, or **neural representations**.

As graphs are discrete set-theoretic structures that typically cannot be input directly to matrix-based machine learning algorithms, representation learning is an especially important component of machine learning on graphs. Graph representation learning approaches map the graph's components (nodes, edges, and/or relation types) to low-dimensional dense vector embeddings using either unsupervised data reconstruction or task-specific supervision objectives. The function f may be as simple as a linear model [Yang et al., 2015] or as complex as a nonlinear multilayer neural network [Kipf and Welling, 2017, Veličković et al., 2018]. Once the representations are learned, it is assumed that they capture the structure and semantics of the input graph within their parameters.

Several types of graph representation learning approach exist [Hamilton, 2020]. In this thesis, we will focus on two: Knowledge base embeddings, which are supervised for the link prediction task, and graph neural networks, which are supervised for the node classification task.

2.4.1 Knowledge Graph Embeddings

Knowledge base embeddings, more commonly called **knowledge graph embeddings** (**KGE**s), are supervised for the link prediction ranking task. KGEs are typically implemented as shallow decoder models that learn embeddings of entities and relations, and compose these embeddings with additive or multiplicative vector functions to output link prediction scores. KGEs are optimized such that observed edges in the graph are scored highly compared to negative samples.

KGEs with **additive scoring functions** treat relationship types as translations or rotations between entities in latent space. Prominent additive KGEs include TransE [Bordes et al., 2013] and RotatE [Sun et al., 2019]. As a concrete example of an additive scoring function, TransE assumes that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ for d-dimensional head, relation, and tail embeddings $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$, and scores links with the negative Euclidean distance $-\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$. KGEs with **multiplicative scoring functions** use bilinear models to score triples. Prominent multiplicative KGEs include RESCAL [Nickel et al., 2011], DistMult [Yang et al., 2015], ComplEx [Trouillon et al., 2016], and TuckER [Balazevic et al., 2019a]. ComplEx, one of the most competitive KGEs for link prediction, scores triples with $\text{re}\left(\mathbf{h}^\top \text{diag}(\mathbf{r})\bar{\mathbf{t}}\right)$, where $\bar{\mathbf{t}}$ is the complex conjugate of $\mathbf{t}$ and re denotes the real part of a complex number. Many other KGEs have been proposed, including variants on the prominent examples here, nonlinear models [Dettmers et al., 2018], and encoder-decoder models [Vashishth et al., 2020b]. For further details on these approaches, we refer the reader to relevant surveys [Nickel et al., 2015, Wang et al., 2017, Ji et al., 2020].

2.4.2 Graph Neural Networks

The second category of graph representation learning approach relevant to this thesis is **graph neural networks**. The majority of graph neural networks are based on the **graph convolutional network** (GCN) architecture [Kipf and Welling, 2017], which uses **message passing** to iteratively update nodes' representations with "messages" from their neighboring nodes, supervising on the node classification task [Hamilton, 2020].

Given a node u and its H-dimensional hidden representation $\mathbf{h}_u^{(k)} \in \mathbb{R}^H$ at layer k in a graph neural network, the message passing rule for computing u's hidden representation in layer $k+1$ is computed recursively, as follows:

$$\mathbf{h}_u^{(k+1)} = \textsc{Update}\left(\mathbf{h}_u^{(k)}, \textsc{Aggregate}\left(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}\right)\right).$$

Aggregate represents a differentiable function that aggregates the hidden representations of u's neighbors into a "message," and Update represents a differentiable function that combines node u's current representation with the aggregated message. In the base case $k = 0$, the node represen-

tation $\mathbf{h}_u^0$ is equal to u's input features $\mathbf{x}_u$, which may be preprocessed text features if available or, if not, a one-hot encoding vector. After ℓ rounds of message passing, the node representations $\mathbf{h}_u^{(\ell)}$ are used to output class label predictions $\hat{\mathbf{y}} = \textsc{Readout}(\mathbf{H}^{(\ell)})$ with a readout layer, which takes the form of trainable K-dimensional classification layer for K classes.

Prominent examples of graph neural networks include the original GCN [Kipf and Welling, 2017], graph attention network (GAT) [Veličković et al., 2018], GraphSAGE [Hamilton et al., 2017], and Simple Graph Convolutions (SGC) [Wu et al., 2019]. That said, many graph neural network architectures exist. For more details on existing models, we refer readers to the relevant surveys [Wu et al., 2020b, Zhou et al., 2020a, Hamilton, 2020].

2.5 Text Representation Learning

Similar to graph-structured data, raw text is discrete in nature and must also be converted to numerical vector features in order to be processed by machine learning algorithms. There is a long history of text featurization in natural language processing and information retrieval. One class of text featurization approaches yields **sparse representations** in which feature vectors are high-dimensional but mostly zero. Examples of sparse text representations include bag-of-words (BOW), which assigns unique IDs to all tokens in a corpus and represents each document by the count of token occurrences, and term frequency-inverse document frequency (TF-IDF), which normalizes the occurrences of terms in documents by the number of documents in which each term appears, in order to diminish the importance of common terms.

Whereas BOW and TF-IDF features correspond directly to words, neural **text representation learning** approaches learn *latent* syntactical and semantic features in text, similar to how graph representation learning learns latent structural features in graphs. **Word embedding** approaches like word2vec [Mikolov et al., 2013] and GloVe [Pennington et al., 2014] map each word in a corpus to d-dimensional latent embedding using unsupervised reconstruction objectives. The embeddings output by word2vec have shown to preserve various syntactic and semantic features in text via vector similarity and distance measures [Schnabel et al., 2015].

2.5.1 Contextual Text Representations

Recently, deep **contextual language models** (LMs) have been proposed to improve upon the limitations of word embeddings. Whereas word embeddings assign a single, static embedding to each word in a corpus, LMs learn rich **contextual embeddings** of tokens such that a single token may have different representations depending on the words that surround it [Peters et al., 2018], reflecting the phenomenon of polysemy in language.

Language models are trained with variants of the statistical language modeling objective, which is a conditional probability estimation task that predicts the likelihood of a token in a sequence given all previous tokens in that sequence [Bengio et al., 2003]. The probability of a sequence of tokens $(t_1, \ldots, t_N)$ is computed as the product of the conditional probabilities of individual tokens:

$$p(t_1, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_1, \ldots, t_{k-1}).$$

The model is trained to output the conditional probabilities $p(t_k|t_1, \ldots, t_{k-1})$ by maximizing the log-likelihood of the sequences seen in the training data. Many variants of this task have been proposed, for example predicting masked tokens conditioned on both preceding and subsequent context [Devlin et al., 2019], predicting adjacency relationships between sentences [Devlin et al., 2019], predicting spans of tokens rather than single words [Joshi et al., 2020, Guu et al., 2020], detection of missing tokens in text [Lewis et al., 2020], and reconstructing permuted sentences in a document [Lewis et al., 2020]. These objectives are used to **pretrain** language models over very large text corpora derived from the Web, which initializes their parameters for generic text representation. LMs may be used as-is after pretraining, or else further **fine-tuned** with supervision on downstream language understanding task(s) by stacking additional layers at their output.

Most contextual LMs today are based on the Transformer deep learning architecture [Vaswani et al., 2017]. Transformers capture long-range dependencies and interactions in sequences by learning **self-attention** weights between all pairs of positions in a sequence. Because Transformers are pairwise models, they can be viewed as learning the edge weights of a fully-connected graph in which each node corresponds to a word in the input sequence. Prominent examples of pretrained LMs based on the Transformer architecture include BERT [Devlin et al., 2019], RoBERTa [Liu et al., 2019], GPT [Brown et al., 2020], T5 [Raffel et al., 2020], and BART [Lewis et al., 2020]. In this thesis, we will focus on BERT and its derivatives. For further details on pretrained language models, we refer readers to the relevant surveys [Qiu et al., 2020, Rogers et al., 2021].

2.5.2 BERT Language Model

The **BERT** (Bidirectional Encoder Representations from Transformers) language model consists of a stack of Transformer encoder layers, each of which computes **multi-headed self-attention** between all pairs of positions in a sequence. A single self-attention head computes attention weights between all pairs of input vectors, which correspond to the representations of tokens in the sequence, and uses these weights to scale the input embeddings. Letting $\mathbf{E} \in \mathbb{R}^{N \times d}$ represent a matrix of d-dimensional token embeddings input to the Transformer layer, self-attention in a Transformer encoder is implemented as $\text{softmax}\left(\frac{\mathbf{E}\mathbf{E}^T}{\sqrt{d}}\right)\mathbf{E}$. A multi-headed attention layer ex-

tends this approach by linearly projecting the input embeddings to different latent subspaces and computing self-attention in parallel in each subspace, then concatenating the outputs and linearly projecting them back to the final desired output dimension. The goal of multi-headed attention is to allow the model to capture different relationships between pairs of inputs in different subspaces, which has empirically shown to improve performance [Vaswani et al., 2017].

In terms of special architectural considerations, BERT treats word inputs by summing three types of word embedding: Token embeddings, positional embeddings that encode the relative position of each token, and segment embeddings that encode one of two possible segments than an input can belong to (i.e., "segment A" or "segment B" in sequence classification tasks). BERT also has two special tokens in its vocabulary: A special **`[CLS]`** token at the beginning of each input sequence that is treated as an aggregate representation of that sequence, and a special **`[SEP]`** token that is used as a delimiter between segments of an input sequence. With BERT, it is common to extract the `[CLS]` token from the last layer as a sequence representation, or else feed it into an additional fully-connected layer that will be fine-tuned for a downstream task. That said, other approaches for extracting semantic text embeddings from BERT have also been proposed, for example averaging the individual token embeddings output by BERT or taking their max per dimension [Reimers and Gurevych, 2019].

BERT is pretrained over English Wikipedia and BooksCorpus [Zhu et al., 2015], which in combination total over three billion words, with two self-supervised objectives: **(1) Masked language modeling**, a reconstruction objective that trains the model to predict randomly masked tokens in the input corpus; and **(2) Next sentence prediction**, a contrastive objective that trains the model to predict whether two sequences are adjacent in the corpus. Note that the masked language modeling objective, by randomly masking tokens in the corpus, allows BERT to condition its token predictions using both preceding and following textual context, making it a **bidirectional** language model. By contrast, **unidirectional** language models are trained to predict the next word in a given sequence, meaning that they can only condition their predictions on preceding context [Bengio et al., 2003, Radford et al., 2018].

From the perspective of text-augmented graph learning, BERT is an attractive choice of language representation for several reasons. In terms of expressiveness, BERT's ability to model bidirectional context and its powerful masked language modeling objective have demonstrated significant advantages over other contextual language models in terms of downstream performance in language understanding tasks [Devlin et al., 2019]. Indeed, BERT has inspired a host of successors, including LMs with the same architecture pretrained on different corpora, as well as LMs with related bidirectional architectures and language reconstruction pretraining objectives [Qiu et al., 2020, Rogers et al., 2021]. In terms of practical utility, BERT's encoder-only structure makes it easy to apply to various downstream tasks by simply stacking an output classification or regression

layer corresponding to the task of interest. Moreover, there are well-established open-source implementations of BERT and its derivatives [Wolf et al., 2020]. Therefore, incorporating a powerful LM like BERT into the domain of graph learning is relatively straightforward for practitioners.

Part I

Relational Knowledge Representation with Language Models

CHAPTER 3

Setting the Stage with a New Taxonomy

The material in this chapter is derived from the paper "Relational World Knowledge Representation in Contextual Language Models: A Review" [Safavi and Koutra, 2021], which appeared in the proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP).

3.1 Introduction

In Chapter 2.3, we introduced relational knowledge bases (**KBs**) as graph data structures that connect pairs of entities or concepts by semantically meaningful symbolic relations. Decades' worth of research have been invested into using KBs as tools for relational world knowledge representation in machines [Minsky, 1974, Lenat, 1995, Liu and Singh, 2004, Bollacker et al., 2008, Vrandečić and Krötzsch, 2014, Speer et al., 2017, Sap et al., 2019, Ilievski et al., 2021].

Most large-scale modern KBs are organized according to a manually engineered schema that specifies which entity and relation types are permitted, and how such types may interact with one another. This explicit enforcement of relational structure is both an advantage and a drawback [Halevy et al., 2003]. On one hand, schemas support complex queries over the data with accurate, consistent, and interpretable answers. On the other hand, schemas are "ontological commitments" [Davis et al., 1993] that limit flexibility in how knowledge is stored, expressed, and accessed. Handcrafted schemas also require significant human engineering effort to construct and maintain, and are therefore often highly incomplete, one of the major limitations of structured knowledge representations [Weikum et al., 2021].

In response to these drawbacks, various research directions in natural language processing (NLP) have proposed approaches to reduce the amount of schema design and manual labor necessary for machine knowledge representation [Halevy et al., 2003, Banko and Etzioni, 2008, Mintz et al., 2009, Fader et al., 2011]. Recently, an especially promising solution has emerged, brought about by breakthroughs in machine learning software, hardware, and data. Specifically, deep con-

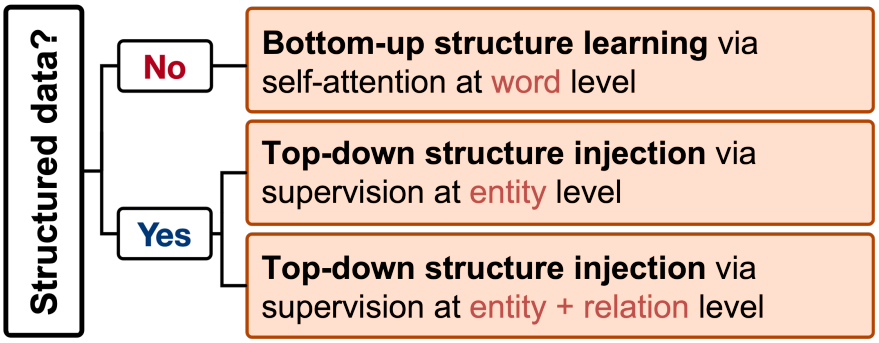


Figure 3.1: A high-level overview of our taxonomy. We organize knowledge representation strategies in language models (LMs) by level of knowledge base (KB) supervision provided to the LM.

textual language models (**LMs**) like BERT [Devlin et al., 2019] and GPT-3 [Brown et al., 2020], which we introduced in Chapters 2.5 and 2.5.2, have shown to be capable of internalizing a degree of KB-like "knowledge" within their parameters, and expressing this knowledge across various mediums and tasks—in some cases, *without* the need for any predefined knowledge representation schema [Petroni et al., 2019, Roberts et al., 2020]. Consequently, some have begun to wonder whether LMs will partially or even fully replace KBs, given sufficiently large training budgets and parameter capacities. To illustrate how LMs may fulfill similar functions to KBs, albeit with very different representation and knowledge acquisition mechanisms, we provide a qualitative comparison in Table 3.1.

3.1.1 Contributions

In this chapter, we introduce a new resource to guide research at the intersection of knowledge bases and language models. Specifically, we propose a taxonomy that organizes relational knowledge representation strategies in LMs by the level of *KB supervision* provided to LMs (Figure 3.1):

- **Word-level supervision** (Ch. 3.2): At this level, LMs are not explicitly supervised on a KB, but may be indirectly exposed to KB-like knowledge via word associations in the training corpus. Here, we cover techniques for probing and utilizing this implicitly acquired knowledge, using KBs as gold-standard evaluation resources.

- **Entity-level supervision** (Ch. 3.3): At this level, LMs are supervised to acquire knowledge of KB entities. Here, we organize entity supervision strategies from "less symbolic" to "more symbolic": Less symbolic approaches train LMs with entity-aware language modeling losses, but never explicitly require the LM to link entity mentions to the KB. By contrast, more symbolic approaches involve entity linking, and may also integrate entity embeddings into the LM's parameters.

Table 3.1: A qualitative comparison of relational knowledge bases and contextual LMs as world knowledge representations.

	Knowledge base (KB)	Contextual language model (LM)
Organization	Graphical data structure that connects pairs of symbolic entities via symbolic relations according to a predefined schema	Matrices of parameter weights learned (optimized) via pretraining and possibly fine-tuning, organized according to a neural architecture
Knowledge acquisition	Explicit population via manual curation and/or automatic information extraction	Implicit (self-supervised pretraining); possibly explicit (supervised fine-tuning)
Knowledge accuracy	Human-curated KBs (Wikidata, ConceptNet) are relatively high-precision but low-recall	Depends on several factors, including number of parameters, amount of training, training strategies, etc; empirical and theoretical limits are not well-understood
Query input	Structured, disambiguated queries potentially mapped from natural language	Arbitrary non-disambiguated sequences, e.g., natural language prefixes, prompts, or questions
Query output	Symbolic entities, relations, and/or structures	Tokens, sequences, and/or labels

- **Relation-level supervision** (Ch. 3.4): At this level, LMs are supervised to acquire knowledge of KB triples and paths. Again, we organize strategies from less to more symbolic, where less symbolic approaches treat triples as fully natural language statements, and more symbolic approaches incorporate dedicated embeddings of KB relation types.

Given the nascence of this research area, our taxonomy is a timely and important contribution to the field. Indeed, the only topically related survey of which we are aware is comparatively narrow in scope and discussion [Colon-Hernandez et al., 2021]. Specifically, our contributions are as follows:

- **New taxonomy**: We introduce a new taxonomy that organizes strategies for relational knowledge representation in pretrained LMs. To illustrate different methods of KB-level input and output in LMs, we order our taxonomy by the level of KB supervision provided to the LM.
- **Technical details and analysis**: We highlight notable methodologies and findings, illustrate concrete technical details via figures, and provide an outlook for future research. Where applicable, we also highlight connections to key questions and tasks in graph learning, in particular within the context of link prediction for automatic knowledge base completion.
- **Guiding vision**: We provide suggestions and outline open questions for future research in this direction.

3.2 Word-Level Supervision

As introduced in Chapter 2.5, the standard language modeling task is to predict the n-th word in a sequence of n words—that is, a conditional probability estimation task [Bengio et al., 2003, Radford et al., 2019]. While many variants of this task have been proposed to allow LMs to condition their predictions on different inputs [Devlin et al., 2019, Raffel et al., 2020, Lewis et al., 2020], a notable feature of all such approaches is that they operate at the word level.

If these supervision techniques do not incorporate KBs at all, how are they relevant when considering LMs as relational knowledge representations? The answer is simple. Typical language modeling corpora like Wikipedia are known to contain KB-like assertions about the world [Da and Kasai, 2019]. LMs trained on enough such data can be expected to acquire some KB-like knowledge, even without targeted entity- or relation-level supervision. Therefore, in order to motivate the necessity (if at all) of KB supervision, it is crucial to first understand what relational world "knowledge" LMs acquire from word-level pretraining alone.

In this section, we cover strategies to extract and utilize this knowledge under the cloze prompting (Ch. 3.2.1) and statement scoring (Ch. 3.2.2) protocols. Table 3.2 provides a taxonomy for this section, with representative examples and evaluation tasks.

3.2.1 Cloze Prompting

The cloze prompting protocol [Taylor, 1953] is a direct approach for extracting and evaluating KB-like knowledge in pretrained LMs. Under this protocol (Figure 3.2), KB triples are first converted to natural language assertions using (e.g.) relation templates. For each assertion, the token(s) corresponding to the object entity are held out. A frozen pretrained LM then ranks candidate tokens within its vocabulary by the probability that they fill in the empty slot(s). Accuracy is typically measured by the proportion of prompts for which the correct answer appears in the LM's top-k predictions, with the assumption that better performance implies more pretrained knowledge within the LM.

Handcrafted prompts in English with single-token answers make up LAMA [Petroni et al., 2019], one of the earliest and most widely-used LM cloze probes. LAMA, which is mapped primarily to Wikidata and ConceptNet triples, was initially used to compare pretrained LMs' knowledge to off-the-shelf KB question answering systems. Petroni et al. [2019] showed that pretrained BERT is competitive with a supervised relation extraction model that has been provided an oracle for entity linking, particularly for 1-1 queries. Subsequent work has experimented with handcrafted templates for probing the knowledge of both very large (hundred-billion parameter) LMs [Brown et al., 2020] as well as non-contextual word embeddings, i.e., as a simple control baseline for

Table 3.2: Taxonomy and representative examples for extracting relational knowledge in word-level pretrained LMs, with evaluation tasks that have been conducted in the referenced papers. *Glossary of evaluation tasks*: KP—knowledge probing; QA—question answering; CR—compositional reasoning; KBC—knowledge base construction.

Probing strategy	**Extraction strategy**	**Representative examples**	**Evaluation task(s)**			
			KP	QA	CR	KBC
Cloze prompts	Prompt handcrafting	Petroni et al. [2019], Dufter et al. [2021]	✓			
	Automatic prompt engineering	Jiang et al. [2020b], Shin et al. [2020], Zhong et al. [2021], Qin and Eisner [2021]	✓			
	Adversarial prompt modification	Kassner and Schütze [2020], Petroni et al. [2020], Poerner et al. [2020], Cao et al. [2021]	✓			
	Varying base prompts	Elazar et al. [2021], Heinzerling and Inui [2021], Jiang et al. [2020a], Kassner et al. [2021]	✓			
	Symbolic rule-based prompting	Kassner et al. [2020], Talmor et al. [2020a]	✓		✓	
Statement scores	Single-LM scoring	Tamborrino et al. [2020], Zhou et al. [2020b]		✓	✓	
	Dual-LM scoring	Davison et al. [2019], Shwartz et al. [2020]		✓		✓

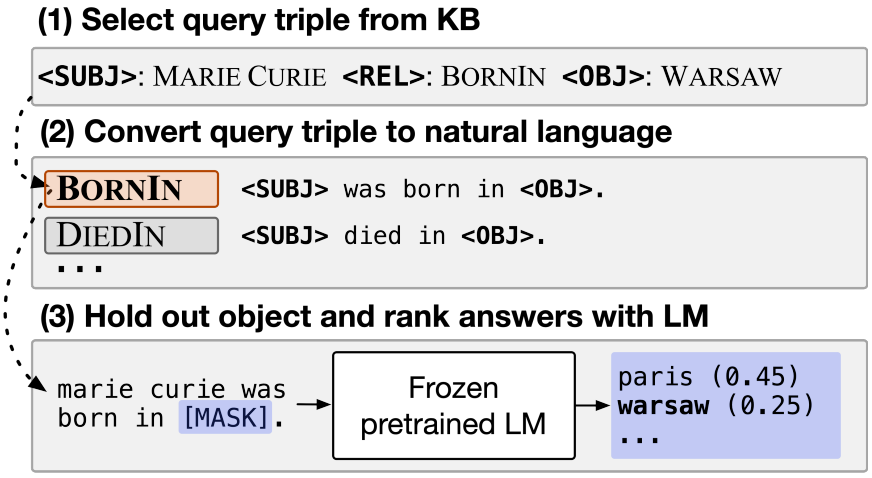


Figure 3.2: Probing relational knowledge in pretrained LMs with cloze prompts generated from KB triples.

LMs [Dufter et al., 2021]. Both studies demonstrate some success, particularly in cases where the probed model is provided a small amount of extra context in the form of conditioning examples [Brown et al., 2020] or entity type information [Dufter et al., 2021].

Automatic prompt engineering is a promising alternative to prompt handcrafting for knowledge extraction in LMs [Liu et al., 2021a], as prompts engineered using discrete [Jiang et al., 2020b, Shin et al., 2020, Haviv et al., 2021] and continuous [Zhong et al., 2021, Qin and Eisner, 2021, Liu et al., 2021b] optimization have improved LMs' lower-bound performance on LAMA's underlying queries. Note, however, that optimized prompts are not always grammatical or intelligible [Shin

et al., 2020]. Prompt optimization methods may also confound knowledge probes by overfitting to the probes' answer distributions during training [Zhong et al., 2021, Cao et al., 2021], and often require large validation sets for tuning, which may not be feasible in practice [Perez et al., 2021].

Adversarial modification of LAMA prompts has uncovered weaknesses in pretrained LMs' world "knowledge," for example that BERT's accuracy drops precipitously when irrelevant statements or negation words are added to prompts [Kassner and Schütze, 2020, Lin et al., 2020, Petroni et al., 2020], and that it can "guess" answers using shallow lexical cues or benchmark artifacts [Poerner et al., 2020, Cao et al., 2021]. However, the adversarial robustness of LM knowledge improves greatly with supervision in both the pretraining [Petroni et al., 2020] and fine-tuning [Kassner and Schütze, 2020] stages, suggesting that explicit KB-level supervision is a viable remedy to input sensitivity.

Several collections of prompt variations, including paraphrased sets of base prompts [Elazar et al., 2021, Heinzerling and Inui, 2021] and multilingual sets of base (English) prompts [Jiang et al., 2020a, Kassner et al., 2021] have been released to expand the original research questions posed by LAMA. For the former, it has been found that pretrained BERT-based LMs typically do not output consistent answers for prompt paraphrases, although their consistency can again be greatly improved by targeted pretraining [Elazar et al., 2021, Heinzerling and Inui, 2021]. For the latter, initial results on prompts beyond English indicate high variability in pretrained LM performance across languages and poor performance on prompts with multi-token answers [Jiang et al., 2020a, Kassner et al., 2021].

Prompts generated with symbolic rules have been used to test pretrained LMs' abilities to learn, e.g., equivalence, implication, composition, and conjunction. Existing studies vary the degrees of experimental control: Talmor et al. [2020a] use BERT-based models with their publicly-available pretrained weights, whereas Kassner et al. [2020] pretrain BERT from scratch on synthetic KB triples only. Both studies observe mixed results, concluding that word-level pretraining alone (at least on BERT) does not lead to strong "reasoning" skills.

3.2.2 Statement Scoring

Beyond probing, pretrained LM "knowledge" can be purposed toward downstream KB-level tasks in a zero-shot manner via statement scoring. Here, a pretrained LM is fed natural language statements corresponding to KB triples, and its token probabilities across each statement are pooled to yield statement scores. These scores are then treated as input to a downstream decision, mirroring the way that supervised LMs can be trained to output probabilities for triple-level prediction tasks (Ch. 3.4). We categorize statement scoring strategies as single- or dual-LM approaches. The

single-LM approach pools the pretrained LM's token scores over a candidate set of sequences, then takes the highest-scoring sequence as the LM's "prediction" or choice [Tamborrino et al., 2020, Bouraoui et al., 2020, Zhou et al., 2020b, Brown et al., 2020]. The **dual-LM** framework first uses one pretrained LM to generate useful context (e.g., clarification text) for the task, then feeds this context to another, possibly different pretrained LM to obtain a final score [Davison et al., 2019, Shwartz et al., 2020]. Notably, the latter approach has shown promise in classification-based link prediction in commonsense KBs, a task that we introduced in Chapter 2.2.1 and will study further in Chapter 4.

Both categories have shown promise over comparable unsupervised (and, under some conditions, supervised) methods for tasks like multiple-choice QA [Tamborrino et al., 2020, Shwartz et al., 2020, Brown et al., 2020] and commonsense KB completion [Davison et al., 2019]. However, LM scores have also shown to be sensitive to small perturbations in text [Zhou et al., 2020b], so this approach may be less effective on noisy or long-tail inputs.

3.2.3 Summary and Outlook

There is still broad disagreement over the nature of acquired "knowledge" in pretrained LMs. Whereas some studies suggest that word-level pretraining may be enough to endow LMs with KB-like knowledge [Petroni et al., 2019, Tamborrino et al., 2020], in particular given enough parameters and the right set of prompts [Brown et al., 2020], others conclude that such pretraining alone does not yield sufficiently precise or robust LM knowledge [Elazar et al., 2021, Cao et al., 2021]—directly motivating the targeted supervision strategies discussed in the remainder of this chapter. We observe that different studies independently set objectives for what a pretrained LM should "know," and thus naturally reach different conclusions.

We believe that future studies must reach consensus on standardized tasks and benchmarks, addressing questions like: What degree of overlap between a pretraining corpus and a knowledge probe is permissible, and how can this be accurately uncovered and quantified? What lexical cues or correlations should be allowed in knowledge probes? Progress in this direction will not only further our understanding of the effects of word-level supervision on LM knowledge acquisition, but will also provide appropriate yardsticks for measuring the benefits of targeted entity- and relation-level supervision.

3.3 Entity-Level Supervision

We next review entity-level supervision strategies for LMs, most often toward improving performance in knowledge probing benchmarks and canonical NLP tasks like entity typing, entity link-

ing, and question answering. We roughly categorize approaches from "least symbolic" to "most symbolic." On the former end of the spectrum, the LM is exposed to entity mentions in text but not required to link these mentions to an external entity bank (Ch. 3.3.1). On the latter end, the LM is trained to link mentions to the KB using late (Ch. 3.3.2) or mid-to-early fusion approaches (Ch. 3.3.3). Table 3.3 provides a taxonomy of supervision strategies for this section with representative examples.

3.3.1 Modeling entities without linking

The "least symbolic" entity supervision approaches that we consider input textual contexts containing entity mention-spans to the LM, and incorporate these mention-spans into their losses. However, they do not require the LM to link these mentions to the KB's entity set, so the LM is never directly exposed to the KB. Figures 3.3a and 3.3b provide examples of input and output for this class of approaches.

Masking tokens in mention-spans and training LMs to predict these tokens may promote knowledge memorization [Sun et al., 2020]. Roberts et al. [2020] investigate this strategy using a simple masking strategy whereby an LM is trained to predict the tokens comprising named entities and dates in text (Figure 3.3a, originally proposed by Guu et al., 2020). The authors find that the largest (11 billion parameter) version of T5 generates exact-match answers on open-domain question answering (QA) benchmarks with higher accuracy than extractive systems—even without access to external context documents, simulating a "closed-book" exam.

Contrastive learning techniques, which have been used for LM supervision at the word and sentence level [Devlin et al., 2019], have also been devised for supervision on entity mentions [Shen et al., 2020]. For example, Xiong et al. [2020b] replace a proportion of entity mentions in the pretraining corpus with the names of negatively-sampled entities of the same type, and train an LM to predict whether the entity in the span has been replaced (Figure 3.3b). Although the previously discussed closed-book T5 model [Roberts et al., 2020] outperforms Xiong et al. [2020b]'s open-book BERT pretrained with contrastive entity replacement on open-domain QA, the latter may generalize better: T5's performance degrades considerably for facts not observed during training, whereas open-book approaches appear more robust [Lewis et al., 2021].

3.3.2 Linking with Late Fusion

The next-strongest level of entity supervision is to train the LM to link entity-centric textual contexts to a KB's entity set E. Here, we cover late fusion approaches, which operate at the word level in terms of input to the LM and incorporate entities at the LM's output layer only, as exem-

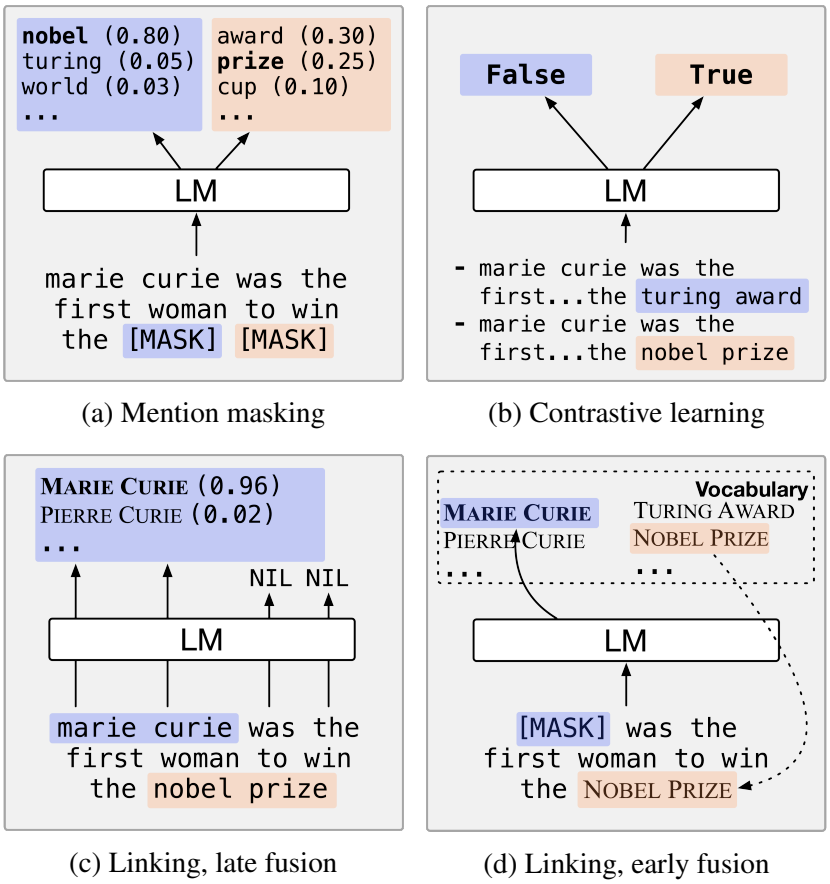
Table 3.3: Taxonomy and representative examples of entity-level supervision in LMs, with evaluation tasks that have been conducted in the referenced papers. *Glossary of evaluation tasks*: KP—knowledge probing; EL—entity linking; ET—entity typing; RC—relation classification; QA—question answering; GL—the General Language Understanding Evaluation or GLUE benchmark [Wang et al., 2019a], which covers multiple subtasks.

Entities as...	**Supervision strategy**	**Representative examples**	**Evaluation task(s)**					
			KP	EL	ET	RC	QA	GL
Token mention-spans	Masked token prediction	[Roberts et al., 2020, Guu et al., 2020]					✓	
	Contrastive learning	[Xiong et al., 2020b, Shen et al., 2020]	✓		✓		✓	
KB links, late fusion	Linking w/o external info	[Broscheit, 2019, Ling et al., 2020]		✓				✓
	Linking w/ textual metadata	[Wu et al., 2020a, De Cao et al., 2021]		✓		✓	✓	
	Linking w/ external embeddings	[Zhang et al., 2019b, Chen et al., 2020a]		✓	✓	✓		✓
KB links, mid/early fusion	Entity embedding retrieval	[Peters et al., 2019, Févry et al., 2020]	✓	✓	✓	✓	✓	
	Treating entities as tokens	[Yamada et al., 2020, Poerner et al., 2020]	✓	✓	✓	✓	✓	

plified in Figure 3.3c. The simplest representatives of this category train LMs to match individual tokens [Broscheit, 2019] or mentions [Ling et al., 2020] in a text corpus to an entity bank, without any external resources. The minimally "entity-aware" BERT proposed by Broscheit [2019], which adds a single classification layer on top of a pretrained BERT encoder, achieves competitive results with a state-of-the-art specialized entity linking architecture [Kolitsas et al., 2018].

Entity meta-information such as names and descriptions are viable external resources for LM-powered entity linking [Botha et al., 2020]. For example, in zero-shot entity linking [Logeswaran et al., 2019], textual mentions must be linked to entities unseen during training using only entity descriptions as additional data. Here, competitive solutions train separate BERT models to select and rank candidate entities by encoding their descriptions [Logeswaran et al., 2019, Wu et al., 2020a]. More recently, encoder-decoder LMs have been trained to retrieve entities by generating their unique names [De Cao et al., 2021], which has the advantage of scaling with the LM's vocabulary size (usually tens of thousands) instead of the KB entity set size (potentially tens of millions). De Cao et al. [2021] achieve results competitive to discriminative approaches on entity linking and QA, suggesting the potential of generative entity-aware LMs.

External entity embeddings pretrained by a separate model have been used as strong sources of inductive bias for LMs. For example, several variants of BERT further pretrain the base model by linearly fusing external entity embeddings with contextual word representations at the output of the BERT encoder [Zhang et al., 2019b, He et al., 2020]. BERT has also been fine-tuned to match its output token representations to external entity embeddings for the task of end-to-end entity linking [Chen et al., 2020a]. Such approaches rely heavily on the quality of the externally-learned embeddings, which is both a strength and a drawback: Such embeddings may contain useful implicit structural information about the KB, but on the other hand may propagate errors into the LM [Shen et al., 2020].



(a) Mention masking

(b) Contrastive learning

(c) Linking, late fusion

(d) Linking, early fusion

Figure 3.3: Examples of entity-level supervision in LMs, ranging from "less symbolic" to "more symbolic."

3.3.3 Linking with Middle or Early Fusion

The last and strongest category of entity supervision techniques that we consider are also linking-based, but fuse entity information at earlier stages of text encoding. Mid-fusion approaches retrieve external entity representations in between hidden layers and re-contextualize them into the LM, whereas early fusion approaches simply treat entity symbols as tokens in the vocabulary. Figure 3.3d provides an example of input/output for early fusion.

Retrieving entity embeddings and integrating them into an LM's hidden word representations is a middle-fusion technique that has the advantage of modeling flexibility: It allows the practitioner to choose where (i.e., at which layer) the entity embeddings are integrated, and how the entity embeddings are learned and re-contextualized into the LM. Peters et al. [2019] integrate externally pre-trained, frozen entity embeddings into BERT's final hidden layers using a word-to-entity attention mechanism. Févry et al. [2020] learn the external entity embeddings jointly during pre-training, and perform the integration in BERT's earlier hidden layers using an attention-weighted

sum. The latter approach is competitive with a 30× larger T5 LM in closed-book QA (Ch. 3.3.1), suggesting that LMs and KB embeddings can be trained jointly to enhance and complement each other.

Treating entities as "tokens" by appending special reserved entity symbols to the LM's vocabulary is the earliest of entity fusion approaches (Figure 3.3d). For instance, Yamada et al. [2020] input entity "tokens" alongside textual contexts that mention these entities to RoBERTa, and use specialized word-to-entity and entity-to-entity attention matrices within its hidden layers. Other approaches leave the base LM's internal architecture completely unchanged and focus only on aligning the LM's word and entity embedding spaces at the input level [Rosset et al., 2020, Poerner et al., 2020]. Note, however, that this approach may significantly enlarge the LM's vocabulary. For example, plain BERT's vocabulary is around 30k tokens, whereas English Wikipedia has around 6 million entities. This can make pretraining on a larger vocabulary expensive in terms of both time and memory usage [Yamada et al., 2020, Dufter et al., 2021].

3.3.4 Summary and Outlook

The literature on entity supervision in LMs is growing rapidly. In line with recent trends in NLP [Khashabi et al., 2020], a growing number of entity supervision strategies use generative models [Roberts et al., 2020, De Cao et al., 2021], which are attractive because they allow for a high level of flexibility in output and circumvent the need for classification over potentially millions of entities. However, some studies find that generative models currently do not perform well beyond what they have memorized from the training set [Wang et al., 2021b, Lewis et al., 2021]. These findings suggest that storing some entity knowledge externally (e.g., in a dense memory, Févry et al., 2020) may be more robust, for example by allowing for efficient updates to the LM's knowledge [Verga et al., 2020]. We believe that future work will need to analyze the tradeoffs between fully-parametric and retrieval-based entity modeling in terms of pure accuracy, parameter and training efficiency, and ability to generalize beyond the training set.

3.4 Relation-Level Supervision

Finally, we consider methods that utilize KB triples or paths to supervise LMs for complex, often compositional tasks like relation classification, text generation, and rule-based inference. We again organize methods in the order of less to more symbolic. In this context, less symbolic approaches treat triples and paths as fully natural language (Ch. 3.4.1 and 3.4.2). By contrast, more symbolic approaches learn distinct embeddings for relation types in the KB (Ch. 3.4.3). Table 3.4 provides a taxonomy of this section with representative examples and evaluation tasks. Note that the methods

Table 3.4: Taxonomy and representative examples of relation-level supervision in LMs, with evaluation tasks conducted in the respective referenced papers. *Glossary of evaluation tasks*: KP—knowledge probing; ET—entity typing; RC—relation classification; QA—question answering; CR—compositional reasoning; KBC—knowledge base construction; TG—text generation; GL—the GLUE family of language tasks [Wang et al., 2019a].

Relations as...	**Supervision strategy**	**Representative examples**	**Evaluation task(s)**							
			KP	ET	RC	QA	CR	KBC	TG	GL
Templated sentences	Lexicalizing triples	[Thorne et al., 2021, Guan et al., 2020]				✓	✓		✓	
	Lexicalizing paths	[Clark et al., 2020, Talmor et al., 2020a,b]	✓				✓			
Linearized sequences	Training on triple sequences	[Yao et al., 2019, Agarwal et al., 2021]	✓			✓		✓	✓	
	Injecting triples into text	[Liu et al., 2020b]				✓				
Dedicated embeddings	Pooling entity representations	[Baldini Soares et al., 2019, Qin et al., 2021]		✓	✓	✓				
	Embedding relations externally	[Wang et al., 2021d, Daza et al., 2021]		✓	✓			✓		✓
	Treating relations as tokens	[Bosselut et al., 2019, Hwang et al., 2021]						✓		

in this section are the most related to graph learning, as several of the papers we review in Ch. 3.4.2 and Ch. 3.4.3 consider variants of the link prediction task in multi-relational graphs.

3.4.1 Relations as Templated Assertions

Template-based lexicalization is a popular relation supervision strategy that does not directly expose the LM to the KB. Similar to how KB queries are converted to cloze prompts for knowledge probing (Ch. 3.2.1), triples are first converted to natural language assertions using relation templates, usually handcrafted. These assertions are then fed as input to the LM, which is trained with any number of task-specific losses. Figure 3.4 provides an input/output example for this class of approach.

Lexicalized triples from Wikidata have been used as LM training data in proof-of-concept studies demonstrating that LMs can serve as natural language querying interfaces to KBs under controlled conditions [Heinzerling and Inui, 2021]. A promising approach in this direction uses encoder-decoder LMs to generate answer sets to natural language queries over lexicalized Wikidata triples [Thorne et al., 2020, 2021], toward handling multi-answer KB queries with LMs—thus far an understudied task in the LM knowledge querying literature.

Other approaches convert KB triples to sentences using relation templates in order to construct task-specific training datasets for improved performance in, e.g., story generation [Guan et al., 2020], commonsense QA [Ye et al., 2020, Ma et al., 2021a], and relation classification [Bouraoui et al., 2020]. While most of these approaches rely on template handcrafting, a few automatically mine templates using distant supervision on Wikipedia, achieving competitive results in tasks like relation classification [Bouraoui et al., 2020] and commonsense QA [Ye et al., 2020].

Compositional paths spanning multiple atoms of symbolic knowledge may also be lexicalized

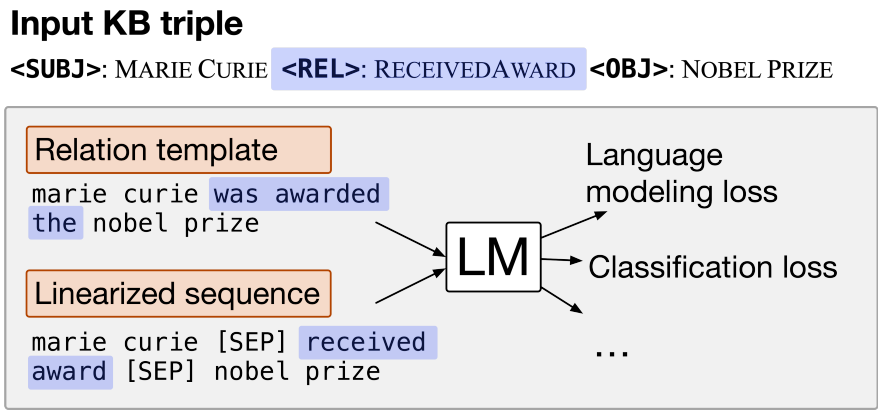


Figure 3.4: Strategies for representing relations as sequences: Templating (Ch. 3.4.1) and linearization (Ch. 3.4.2).

and input to an LM [Lauscher et al., 2020, Talmor et al., 2020a] in order to train LMs for soft compositional reasoning [Clark et al., 2020, Talmor et al., 2020b]. Notably, when RoBERTa is fine-tuned on sentences expressing (real or synthetic) facts and rules from a KB, it can answer entailment queries with high accuracy [Clark et al., 2020, Talmor et al., 2020b]. However, as Clark et al. [2020] note, these results do not necessarily confirm that LMs can "reason," but rather that they can at least emulate soft reasoning—raising an open question about how to develop probes and metrics to verify whether LMs can actually reason compositionally.

3.4.2 Linearizing KB Triples

The main advantage of templating is that it converts symbolic triples into sequences, which can be straightforwardly input to LMs. However, handcrafting templates is a manual process, and distant supervision can be noisy. To maintain the advantage of templates while avoiding the drawbacks, triples can alternatively be fed to an LM by linearizing them—that is, flattening the subject, relation, and object into an input sequence (Figure 3.4). With linearization, relation-level supervision becomes as simple as **feeding the linearized sequences** to the LM and training again with task-specific losses [Yao et al., 2019, Kim et al., 2020, Ribeiro et al., 2021, Wang et al., 2021a] or **injecting the sequences into the pretraining corpus** [Liu et al., 2020b]. A notable recent example of the former approach [Agarwal et al., 2021] trains T5 on linearized Wikidata triples in order to generate fully natural language versions of those triples. These verbalized triples are used as retrieval "documents" for improved LM-based QA over traditional document corpora; note, however, that they can also be used as LM training data for other downstream tasks in place of handcrafted templates (Ch. 3.4.1).

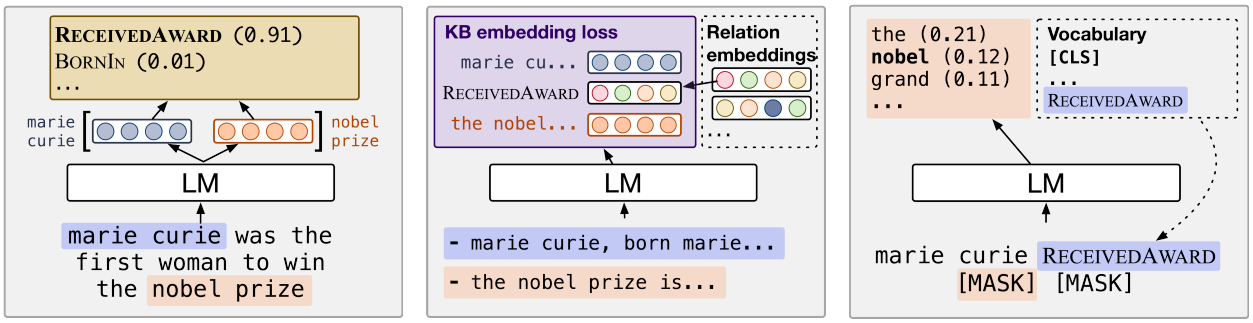
3.4.3 Relations as Dedicated Embeddings

The strategies discussed thus far treat KB triples and paths as natural language sequences. A "more symbolic" approach is to represent KB relation types with dedicated embeddings, and integrate these embeddings into the LM using late, middle, or early fusion approaches. Figures 3.5a and 3.5b provide input/output examples for late fusion, whereby relation textual contexts are input to the LM, and relation embeddings are constructed or integrated at the LM's output. Figure 3.5c exemplifies early fusion, whereby relations are treated as input tokens.

Contextual representations of entity mention-spans may be pooled at an LM's output layer to represent a relation [Wang et al., 2021c, Yu et al., 2020]. For example, Baldini Soares et al. [2019] concatenate the contextual representations of special entity-start markers inserted adjacent to textual entity mentions, and fine-tune BERT to output similar relation representations for statements ranging over the same entity pairs (Figure 3.5a). This approach, which proved highly successful for relation classification, has been applied to the same task in languages beyond English [Köksal and Özgür, 2020, Ananthram et al., 2020], and as an additional LM pretraining objective [Qin et al., 2021].

Non-contextual relation embeddings may be learned by defining a separate relation embedding matrix with $|R|$ rows and fusing this matrix into the LM. One advantage of this approach, similar to methods for retrieving external entity embeddings (Ch. 3.3.3), is that it supports fusion at both the late [Wang et al., 2021d, Daza et al., 2021] and middle [Liu et al., 2021c] stages. As an example of the former, Wang et al. [2021d] propose an LM pretraining objective whereby textual descriptions of KB entities are input to and encoded by an LM, then combined with externally-learned relation embeddings at the output using a link prediction loss (Figure 3.5b). Combined with standard word-level language modeling objectives, this approach enables generalization across both sentence-level tasks like relation classification, and graph-level tasks like KB completion.

Treating relations as "tokens," toward early fusion of relations in LMs, is achieved by appending the KB's relation types to the LM's vocabulary (Figure 3.5c). A notable instantiation of this approach is the COMET commonsense KB construction framework, which aims to augment a semi-structured commonsense KB (Chapter 2.3) with novel edges [Bosselut et al., 2019, Hwang et al., 2021, Jiang et al., 2021]. Specifically, given a subject phrase/relation prompt as input, COMET fine-tunes an LM to generate object phrases. COMET demonstrates promising improvements over 400$\times$ larger LMs not trained for KB construction [Hwang et al., 2021]. However, templating (Ch. 3.4.1) may yield better results than adding special tokens to the vocabulary when the COMET framework is trained and tested in a few-shot setting [Da et al., 2021].



(a) Entity pair pooling (b) Relation embeddings (c) Relations as "tokens"

Figure 3.5: Examples of relation supervision strategies that incorporate dedicated embeddings of relation types.

3.4.4 Summary and Outlook

Relation-level supervision in LMs is exciting because it enables a wide variety of complex NLP tasks (Table 3.4). A unifying theme across many of these tasks is that of compositionality, or the idea that smaller "building blocks" of evidence can be combined to arrive at novel knowledge. As compositionality is thought to be key to machine generalization [Lake et al., 2017], we believe that further fundamental research in understanding and improving LMs' soft "reasoning" skills [Clark et al., 2020, Talmor et al., 2020b] will be crucial (Ch. 3.4.1).

Finally, while most of the open directions we discuss involve improving LM knowledge with KBs, we find the direction of generating KBs with LMs equally intriguing. This direction, which can be formulated as the fundamental graph learning task of link prediction, reflects the fact that LMs and KBs can complement each other in "both directions": That is, inasmuch as KBs provide useful structured training sources for LMs, LMs can also help automate and scale out the construction of higher-quality KBs. The generative COMET framework [Bosselut et al., 2019] and its successors has made inroads in commonsense KB construction (Ch. 3.4.3), but the same progress has not yet been observed for encyclopedic knowledge. The latter entails unique challenges: Whereas commonsense entities are not disambiguated and triples need only be plausible rather than always true, encyclopedic entities are usually disambiguated and facts are often binary true/false. Toward this goal, in Chapter 5 we will discuss novel models and resources for factual KB completion via text-augmented graph learning. We also look forward to future research that addresses this challenge from alternative perspectives, for example via generative factual entity retrieval (Ch. 3.3.2).

3.5 Conclusion

In this chapter, we proposed a novel taxonomy for relational world knowledge representation in language models (LMs). We categorized knowledge representation strategies by the level of knowledge base (KB) supervision provided to an LM, from no explicit supervision at all to entity- and relation-level supervision. Within our taxonomy, we highlighted notable methodologies and findings, illustrated concrete technical details, and made connections where applicable to key tasks in graph learning.

Our review indicates that while LMs may not be viable replacements for KBs yet, LMs are significantly expanding the utility of KBs by providing flexible interfaces to structured knowledge that can be utilized in a host of complex language and reasoning tasks. Our review also highlights the complementary and synergistic aspects of LMs and KBs. As we have shown, the knowledge recall and reasoning abilities of LMs can be improved significantly using KB-level supervision. On the converse, KBs can be automatically generated and augmented with LMs, in order to scale them out without costly manual curation. In the following chapters we will focus on the latter direction, that of using LMs to automatically complete KBs, and show that their pretrained "knowledge" and adaptability to downstream tasks make them ideal for challenging KB completion tasks.

CHAPTER 4

Inferring Negative Commonsense Knowledge

The material in this chapter is derived from the paper "NegatER: Unsupervised Discovery of Negatives in Commonsense Knowledge Bases" [Safavi et al., 2021], which appeared in the proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP).

4.1 Introduction

Having introduced language models (LMs) as powerful tools for machine knowledge representation and reasoning in the previous chapter, we now consider a knowledge acquisition task to which LMs are well-suited. In particular, endowing machines with "commonsense," which is knowledge that members of a culture usually agree upon but do not express explicitly, is a major but elusive goal of artificial intelligence [Minsky, 1974, Davis et al., 1993, Liu and Singh, 2004, Davis and Marcus, 2015]. One way to capture such knowledge is with curated commonsense knowledge bases (KBs), which contain semi-structured statements of "everyday" human knowledge, for example pre-conditions of events, properties of objects, and outcomes of actions. As such KBs are increasingly being used to augment the capabilities of intelligent agents [Hwang et al., 2021], automatically expanding their scope has become crucial [Li et al., 2016, Davison et al., 2019, Bosselut et al., 2019, Malaviya et al., 2020].

Previous research in this direction focuses primarily on the acquisition of positive knowledge, or that which is true about the world. However, understanding what is true about the world often also requires gathering and reasoning over explicitly *untrue* information. Humans routinely rely on **negative knowledge**—that is, what "not to do" or what "not to believe"—in order to increase certainty in decision-making and avoid mistakes and accidents [Minsky, 1994]. Similarly, discriminative models that operate over structured knowledge from KBs often require explicit negative examples in order to learn good decision boundaries [Sun et al., 2019, Ahrabian et al., 2020, Ma et al., 2021a].

Table 4.1: Out-of-KB statements are less meaningful as negative examples when sampled at random versus ranked with our NegatER framework. The random examples are taken from the test split of the ConceptNet benchmark introduced by Li et al. [2016].

Method	Negative statement
Random sampling	(“tickle”, `HasSubevent`, “supermarket”) (“lawn mower”, `AtLocation`, “pantry”) (“closet”, `UsedFor`, “play baseball”)
NegatER ranking	(“ride horse”, `HasSubevent`, “pedal”) (“zoo keeper”, `AtLocation`, “jungle”) (“air ticket”, `UsedFor`, “get onto trolley”)

The main challenge with machine acquisition of structured negative knowledge, commonsense or otherwise, is that most KBs do not contain negatives at all [Arnaout et al., 2020]. Therefore, for KB-related tasks that require both positive and negative statements, negatives must either be gathered via human annotation, or else generated ad-hoc. Both of these approaches entail distinct challenges. On one hand, human annotation of negatives can be cost-prohibitive at scale. On the other, automatic negative generation without good training examples can lead to uninformative, even nonsensical statements (Table 4.1), because the prevailing approach is to randomly sample negatives from the large space of all out-of-KB statements [Li et al., 2016].

4.1.1 Contributions

To strike a balance between expert annotation, which is costly but accurate, and random sampling, which is efficient but inaccurate, in this chapter we propose **NegatER**, a framework for unsupervised discovery of **Negat**ive Commonsense Knowledge in **E**ntity and **R**elation form. Rather than randomly sampling from the space of all out-of-KB statements to obtain negatives, NegatER *ranks* a selection of these statements such that higher-ranking statements are “more likely” to be negative. We propose to rank statements using a fine-tuned contextual language model (LM), building upon our taxonomy in the previous chapter, in which we demonstrated that LMs can be trained with relation-level supervision strategies to express world knowledge.

Importantly, because we do not assume the presence of gold negative examples for training the LM, we devise techniques that make use of *positive KB statements only*. This distinguishes NegatER from supervised generative commonsense KB construction techniques that require abundant gold examples, usually obtained via human annotation, for fine-tuning [Bosselut et al., 2019, Hwang et al., 2021, Jiang et al., 2021]. Our realistic assumption means that we do not have any explicit examples of true negatives and therefore cannot guarantee a minimum true negative rate;

indeed, obtaining true negatives in KBs is a hard problem in general [Arnaout et al., 2020]. However, we show in detailed experiments that NegatER strikes a delicate balance between several factors that contribute to high-quality negative knowledge, including task-specific utility, coherence, and the true negative rate.

Our contributions are as follows:

- **New problem definition** (Ch. 4.3): We provide the first rigorous definition of negative knowledge in commonsense KBs, which as far as we are aware has not been studied before.
- **Unified methodology** (Ch. 4.4): We introduce NegatER, a new approach for negative commonsense knowledge representation and reasoning that combines relational knowledge bases and language models in a unified framework. NegatER ranks out-of-KB potential negatives using a contextual LM to discover negatives. As KBs typically do not contain gold negatives, we devise an approach that relies only on the LM's *positive* beliefs. Specifically, NegatER first fine-tunes the LM to acquire high-quality positive knowledge, then ranks potential negatives by how much they "contradict" the LM's positive knowledge, as measured by its classification scores or gradients.
- **Extensive experiments** (Ch. 4.5, 4.6, and 4.7): In keeping with the novelty of the problem, we conduct multiple experimental evaluations that address the fundamental research questions of negative commonsense. First, we measure the effectiveness of our LM fine-tuning approach and the utility of NegatER-generated negatives in KB completion tasks. Next, we study the intrinsic quality of the generated negatives. When considering all such factors, NegatER outperforms numerous competitive baselines. Most notably, training KB completion models with highly-ranked negative examples from NegatER results in statistically significant accuracy improvements of up to 1.90%.

4.2 Related Work

Commonsense KB completion Existing approaches to automatic commonsense KB completion include link prediction, both classification-based [Li et al., 2016, Saito et al., 2018, Jastrzebski et al., 2018, Davison et al., 2019] and ranking-based [Malaviya et al., 2020], as well as generative approaches that train decoder language models to complete triple prefixes with novel knowledge [Bosselut et al., 2019, Hwang et al., 2021]. Such approaches either focus solely on modeling positive knowledge, or else generate negatives at random, making our work the first attempt at automatically generating meaningful negative knowledge.

Knowledge in language models As discussed in our review of LMs as world knowledge representations (Ch. 3), several studies have shown that deep contextual language models acquire a degree of implicit commonsense "knowledge" during pretraining [Petroni et al., 2019, Davison et al., 2019, Roberts et al., 2020], although this knowledge has shown to be brittle to, for example, linguistic negation [Kassner and Schütze, 2020, Ettinger, 2020]. Note, of course, that we distinguish negation from negative knowledge, as the latter specifically refers to false statements that do not necessarily contain negative particles. Beyond pretraining, there is ample evidence that the accuracy and robustness of LM knowledge can be improved significantly by targeted fine-tuning [Bosselut et al., 2019, Kassner and Schütze, 2020, Jiang et al., 2021, Hwang et al., 2021]. We take the latter direction in this chapter, but toward the novel goal of generating negative rather than positive knowledge.

Negative sampling While we are not aware of any existing work on negative sampling for commonsense knowledge, several negative samplers for encyclopedic KBs like Freebase and Wikidata exist, including self-adversarial [Cai and Wang, 2018, Sun et al., 2019], graph-structural [Ahrabian et al., 2020], and heuristic "interestingness" [Arnaout et al., 2020] approaches. While these methods share our high-level goal, we show in our experiments that they are less effective on highly sparse commonsense KBs.

Beyond knowledge bases, the problem of mining or generating hard training negatives is well-studied [Ying et al., 2018, Cohan et al., 2020, Xiong et al., 2020a, Ma et al., 2021a]. Ying et al. [2018] proposed a hard negative mining strategy for improving the precision of recommender systems, as the class distribution of positive and negative items are typically highly imbalanced in such settings. Xiong et al. [2020a] proposed a self-adversarial negative sampling strategy for improving the discriminative ability of dual encoders for information retrieval, demonstrating large gains in retrieval accuracy with better negative samples. From the commonsense question answering community, several studies have addressed the concept of "distractor" negatives as plausible but incorrect answers to multiple-choice questions [Talmor et al., 2019, Shen et al., 2020, Ma et al., 2021a]. The latter research direction is the closest to ours, but such studies consider question answering, whereas we focus on the graph learning task of KB completion.

4.3 Problem Definition

As the problem of negative knowledge has not yet been addressed in the commonsense KB completion literature, we begin by defining meaningful negatives in commonsense KBs.

Positive knowledge A commonsense knowledge base (KB) consists of triples $\{x^+\} = \{(X_h, r, X_t)^+\}$, where the superscript denotes that all in-KB triples are assumed to be positive or true. As discussed in Chapter 2.3, commonsense KBs are semi-structured and do not disambiguate entities. In each triple, the head and tail entities take the form of free-text phrases $X_h = [w_1, \ldots, w_h]$ and $X_t = [w_1, \ldots, w_t]$ drawn from a potentially infinite vocabulary. The relation types r are symbolic and drawn from a finite dictionary R. Figure 4.1 provides examples of positive statements from the ConceptNet KB [Speer and Havasi, 2012], e.g., (X_h="horse", r=`IsA`, X_t="expensive pet").

Negative knowledge We denote a negative triple as $x^- \notin \{x^+\}$. As the space of negatives is evidently much larger than the space of positives, we define negative knowledge to exclude trivial negatives, for example simple negations or nonsensical statements.

Drawing from the literature on procedural negative expertise in humans [Minsky, 1994, Gartmeier et al., 2008], we define negative knowledge as nonviable or explicitly false knowledge that is heuristically valuable with respect to a given task, goal, or decision. In the context of KBs, we devise three requirements that, combined, satisfy this definition:

R1 Negative knowledge must resemble positive knowledge in structure. This means that negative statements should **obey the grammatical rules** (parts of speech) of their relation types.

R2 The head and tail phrases must be **thematically or topically consistent**. For example, given the head phrase X_h="make coffee," a consistent tail phrase is one that is thematically related but still nonviable with respect to the whole statement, for example ("make coffee", `HasSubevent`, "buy tea").

R3 Negative knowledge must be informative for a given task, goal, or decision. We consider a statement as informative if, when taken as true, it is **counterproductive or contradictory** to the goal at hand, e.g., ("make coffee", `HasSubevent`, "drop mug").

4.4 Methodology

We propose the NegatER framework to solve the problem of negative knowledge defined in the previous section.

4.4.1 NegatER Overview

As shown in Figure 4.1, NegatER consists of two steps:

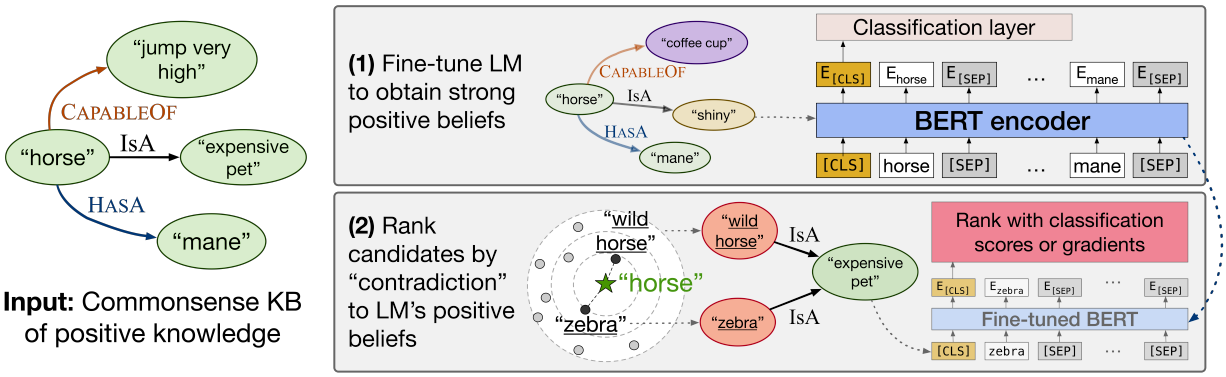


Figure 4.1: NegatER consists of two steps: **(1)** Fine-tuning an LM on the input KB to obtain strong positive beliefs; and **(2)** Feeding a set of out-of-KB candidate statements to the fine-tuned LM and ranking them by the LM's classification scores or gradients. Here, the KB is a fragment of ConceptNet [Speer and Havasi, 2012].

1. A pretrained LM is fine-tuned on a given commonsense KB using a contrastive approach to acquire strong positive beliefs.

2. A set of grammatical (**R1**) and topically consistent (**R2**) out-of-KB candidate statements are fed to the LM and ranked by the degree to which they "contradict" the LM's fine-tuned positive beliefs (**R3**), such that the higher-ranking statements are more likely to be negative.

We emphasize that ground-truth negative examples are *not required* at any point, which means that we trade off some accuracy (i.e., the true negative rate) for cost efficiency (i.e., the cost of gathering ground-truth negative examples for training via expert annotation).

4.4.2 Injecting Positive Knowledge into LMs

The first step of NegatER is to minimally fine-tune a language model on a given commonsense KB using contrastive learning (step 1, Figure 4.1), such that it acquires strong positive beliefs. We focus on encoder-only BERT-based models [Devlin et al., 2019, Liu et al., 2019], which we introduced as powerful contextual text representation tools in Chapter 2.5.2, as we will ultimately use their fine-tuned encodings to represent triples.

LM input and output We input a KB triple (X_h, r, X_t) to the LM by concatenating BERT's special `[CLS]` token with a linearized version of the triple, delineating the head tokens X_h, the relation r, and the tail tokens X_t with BERT's special `[SEP]` token. At the output of the encoder, we apply a semantic-level pooling operation (e.g., any of those proposed by Reimers and Gurevych [2019]) to obtain a single contextual representation of the triple, and pass it through a classification layer $W \in \mathbb{R}^H$, where H is the hidden layer dimension.

Supervision strategy Since the goal of fine-tuning is to endow the LM with strong positive beliefs, we use a common contrastive data augmentation technique for positive KB triple classification [Li et al., 2016, Malaviya et al., 2020]. Specifically, for each positive x^+, we construct a contrastive corrupted version where the head, relation, or tail has been replaced by a random phrase or relation from the KB. We minimize binary cross-entropy loss between the positive training examples and their corrupted counterparts. We learn a decision threshold θ_r per relation r on the validation set to maximize validation accuracy, such that triples of relation r scored above θ_r are classified as positive.

4.4.3 Ranking Out-of-KB Statements

Now that we have an LM fine-tuned to a given commonsense KB, we feed a set of out-of-KB candidate statements to the LM in the same format as was used during fine-tuning, and rank them by the degree to which they "contradict" the LM's positive beliefs (step 2, Figure 4.1).

Out-of-KB candidate generation To gather out-of-KB candidate statements, we use a dense k-nearest-neighbors retrieval approach. The idea here is that the set of *all* out-of-KB statements is extremely large and most such statements are not likely to be meaningful, so we narrow the candidates down to a smaller set that is more likely to be grammatical (**R1**) and consistent (**R2**).

For each positive triple $x^+ = (X_h, r, X_t)^+$, we retrieve the k nearest-neighbor phrases to head phrase X_h using a maximum inner product search [Johnson et al., 2019] over pre-computed embeddings of the KB's entity phrases. While any choice of embedding and distance measure may be used, we use Euclidean distance between the `[CLS]` embeddings output by a separate pre-trained BERT model for its empirical good performance. We then replace X_h in the head slot of the original positive x^+ by each of its neighbors $\tilde{X}_h$ in turn, yielding a set of candidates

$$\{\tilde{x}\}_{i=1}^{k}, \;\; \tilde{x} = (\tilde{X}_h, r, X_t).$$

We discard any candidates that already appear in the KB and repeat this process for the tail phrase X_t, yielding up to $2k$ candidates $\tilde{x}$ per positive x^+. We also filter the candidates to only those for which the retrieved head (tail) phrase $\tilde{X}_h$ ($\tilde{X}_t$) appears in the head (tail) slot of relation r in the KB.

Meeting R1, R2, and R3 Our filtering process discards candidate triples whose head/tail entities have not been observed to co-occur with relation r, which preserves the grammar (**R1**) of the relation. Notice that by retrieving the nearest neighbors of each head and tail phrase by semantic similarity, we also preserve the topical consistency (**R2**) of the original positive statements.

Finally, to meet requirement **R3**, we rank the remaining out-of-KB candidates by the degree to which they "contradict" the positive beliefs of the fine-tuned LM. These ranked statements can be then taken in order of rank descending as input to any discriminative KB reasoning task requiring negative examples, with the exact number of negatives being determined by the practitioner and application. We propose two independent ranking strategies:

4.4.3.1 NEGATER-θ_r: Ranking with Scores

Our first approach, NEGATER-θ_r, relies on the decision thresholds θ_r set during the validation stage of fine-tuning. We feed the candidates $\tilde{x}$ to the LM and take only those that the LM classifies below the respective decision threshold θ_r. Per relation r, the candidates are ranked descending by their scores at the output of the classification layer, such that the higher-ranking candidates look more plausible—that is, "almost positive"—to the LM.

4.4.3.2 NEGATER-∇: Ranking with Gradients

The premise of our second approach, NEGATER-∇, is that the candidates that most "surprise" the LM *when labeled as true* are the most likely to be negative, because they most directly contradict what the LM has observed during fine-tuning.

We quantify "surprisal" with the LM's gradients. Let $\mathcal{L}(\tilde{x}; \tilde{y})$ be the binary cross-entropy loss evaluated on candidate $\tilde{x}$ given a corresponding label $\tilde{y} \in \{-1, 1\}$. We feed each $\tilde{x}$ to the LM and compute the magnitude of the gradient of $\mathcal{L}$ with respect to the LM's parameters Θ, given a positive labeling of $\tilde{x}$:

$$\tilde{M} = \left\| \frac{\partial \mathcal{L}(\tilde{x}; \tilde{y} = 1)}{\partial \Theta} \right\|, \tag{4.1}$$

and rank candidates in descending order of gradient magnitude $\tilde{M}$. Here, $\tilde{M}$ signifies the amount to which the LM's fine-tuned beliefs would need to be updated to incorporate this candidate as positive. Therefore, the higher the $\tilde{M}$, the more directly $\tilde{x}$ contradicts or negates the LM's positive beliefs.

Faster computation Because NEGATER-∇ requires a full forward and backward pass for each candidate $\tilde{x}$, it can be costly for a large number N of candidates. We therefore propose a simple (optional) trick to speed up computation using a "proxy" for the gradients, as demonstrated in Figure 4.2. We first compute $\tilde{M}$ for an initial sample of $n \ll N$ candidates. We then use the contextual representations of these n candidates and their gradient magnitudes $\tilde{M}$ as training features and targets, respectively, to learn a regression function $\tilde{f_M} : \mathbb{R}^H \to \mathbb{R}$. Finally, we substitute

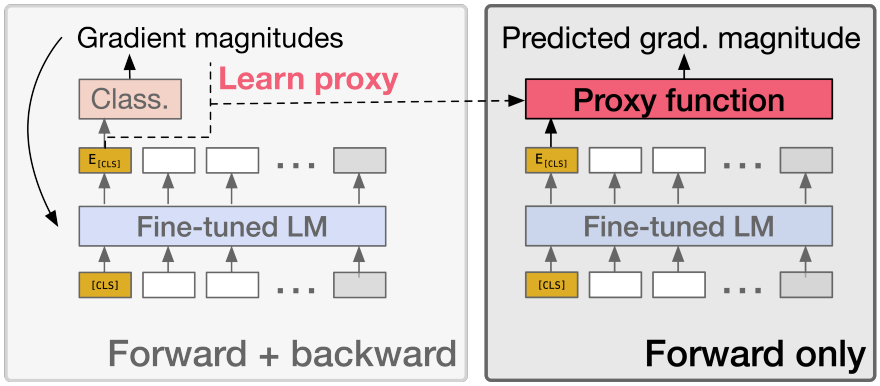


Figure 4.2: Improving the efficiency of NEGATER-∇.

the LM's fine-tuning layer with $\tilde{f}_M$, allowing us to skip the backward pass and feed batches of candidates $\tilde{x}$ to the LM in forward passes. In our experiments, we will show that this approach is an effective and efficient alternative to full-gradient computation.

Gradients versus losses On the surface, it might seem that NEGATER-∇ could be made more efficient by ranking examples descending by their *losses*, instead of gradients. However, notice that the binary cross-entropy loss $\mathcal{L}(\tilde{x}; \tilde{y} = 1)$ is low for candidates $\tilde{x}$ that receives high scores from the LM, and high for candidates that receive low scores. Due to the contrastive approach that we used for fine-tuning, candidates with the lowest losses are mainly *true* statements, and candidates with the highest losses are mainly *nonsensical* statements. Therefore, the losses do not directly correlate with how "contradictory" the candidate statements are. By contrast, the gradient-based approach quantifies how much the LM would need to change its beliefs to incorporate the new knowledge as positive, which more directly matches requirement **R3**.

4.5 Fine-Tuning Evaluation

In this section, we evaluate the efficacy of the fine-tuning step of NegatER. In the following sections, we will evaluate the efficacy of the ranking step of NegatER from quantitative and qualitative perspectives.

4.5.1 Experimental Setup

Data The goal of this experiment is to evaluate whether our fine-tuning strategy from Ch. 4.4.2 endows LMs with sufficiently accurate positive knowledge. For this, we use the dataset introduced by Li et al. [2016], which is a classification-based link prediction benchmark consists of

Table 4.2: Our fine-tuned BERT reaches state-of-the-art accuracy on the ConceptNet benchmark from [Li et al., 2016]. Baseline results are reported directly from the referenced papers.

	Accuracy
Bilinear AVG [Li et al., 2016]	91.70
DNN AVG [Li et al., 2016]	92.00
DNN LSTM [Li et al., 2016]	89.20
DNN AVG + CKBG [Saito et al., 2018]	94.70
Factorized [Jastrzebski et al., 2018]	79.40
Prototypical [Jastrzebski et al., 2018]	89.00
Concatenation [Davison et al., 2019]	68.80
Template [Davison et al., 2019]	72.20
Template + Grammar [Davison et al., 2019]	74.40
Coherency Ranking [Davison et al., 2019]	78.80
KG-BERT$_{\text{BERT-BASE}}$ [Shen et al., 2020]	93.20
KG-BERT$_{\text{GLM(RoBERTa-LARGE)}}$ [Shen et al., 2020]	94.60
Fine-tuned BERT (ours)	**95.42**
Fine-tuned RoBERTa (ours)	94.37
Human estimate [Li et al., 2016]	95.00

100K/2400/2400 train/validation/test triples across 34 relations and 78,334 unique entity phrases from the English-language ConceptNet 5 knowledge base [Speer and Havasi, 2012]. This benchmark has been used widely in the commonsense KB completion literature [Saito et al., 2018, Jastrzebski et al., 2018, Davison et al., 2019, Bosselut et al., 2019, Shen et al., 2020].

Task We consider the original evaluation task proposed by Li et al. [2016] on the ConceptNet dataset, formulated as classification-based KB link prediction (Chapter 2.2.1). The evaluation metric is binary classification accuracy. In the evaluation splits, which are balanced positive/negative 50/50, the negatives were constructed by swapping the head, relation, or tail of each positive x^+ with that of another randomly sampled positive from the KB. The task is technically *inductive* link prediction, as the head/tail entity phrases in the test set triples are not necessarily contained within the train set triples.

Note that while the test negatives were generated randomly and are therefore mostly nonsensical (Table 4.1), we use this benchmark because it mainly tests models' recognition of positive knowledge, which matches the goals of our fine-tuning procedure. Ultimately, however, a more difficult dataset will be needed, which we will introduce in the next section.

Baselines As baselines, we consider all published results on the same ConceptNet evaluation splits of which we are aware. Our baselines include both KB embeddings [Li et al., 2016, Jastrzeb-

ski et al., 2018] and contextual LMs [Davison et al., 2019, Shen et al., 2020].

LM variants We fine-tune BERT-BASE [Devlin et al., 2019] and RoBERTa-BASE [Liu et al., 2019]. To obtain a single contextual representation of a triple from a sequence of triple tokens, we experiment with three standard pooling approaches [Reimers and Gurevych, 2019]: Taking the reserved `[CLS]` token embedding from the output of the encoder, and mean- and max-pooling over all output token representations. As we do not observe statistically significant differences in performance among the pooling operations, we use the `[CLS]` token as the triple embedding, since this is the established approach for textual embedding with BERT..

Software and hardware We implement our LMs with the Transformers PyTorch library [Wolf et al., 2020] and run all experiments on a NVIDIA Tesla V100 GPU with 16 GB of RAM. Both BERT and RoBERTa take around 1.5 hours/epoch to train on the ConceptNet benchmark. We search manually among the following hyperparameters (best configuration for BERT in **bold**, RoBERTa underlined): Batch size in $\{\underline{\mathbf{16}}, 32\}$; Learning rate in $\{10^{-4}, 10^{-5}, \underline{\mathbf{2 \times 10^{-5}}}, 3 \times 10^{-5}\}$; Number of epochs in $\{3, 5, \underline{7}, 10, \mathbf{13}\}$; Number of warmup steps in $\{0, \underline{\mathbf{10K}}, 100\text{K}\}$; Maximum sequence length in $\{16, \underline{\mathbf{32}}, 64\}$. All other hyperparameters are as reported in [Devlin et al., 2019].

4.5.2 Results and Discussion

The results in Table 4.2 confirm the effectiveness of our fine-tuning approach, as our BERT reaches state-of-the-art accuracy on ConceptNet. It even outperforms KG-BERT$_{\text{GLM(RoBERTa-LARGE)}}$ [Shen et al., 2020], which requires an entity linking step during preprocessing and uses a RoBERTa-LARGE model pretrained with several extra tasks. In fact, we suspect that our fine-tuned BERT has saturated this benchmark, as it slightly exceeds the human accuracy estimate provided by Li et al. [2016]. This motivates us to use a harder evaluation set in our next experiments.

4.6 Task-Based Evaluation

We next evaluate the efficacy of the ranking step in NegatER. Specifically, we next show how the top-ranking negative examples from NegatER can be informative (**R3**) for training KB completion models. Similar to the previous section, we fine-tune pretrained BERT and RoBERTa models for a commonsense triple classification task. However, here we use a more challenging dataset split, and vary the ways that negatives are sampled at training time.

4.6.1 Experimental Setup

Data As discussed previously, the ConceptNet split introduced by Li et al. [2016] is already saturated by BERT, likely because it contains "easy" negative test examples. We therefore construct a new, more challenging link prediction split by taking the small percentage (3%) of triples in the benchmark with negated relations (e.g., `NotIsA`, six total), each of which has a positive counterpart in the KB (e.g., `IsA`). We filter the dataset to the positive/negated relation pairs only, and take the negated triples as *true negative instances* for testing by removing the `Not-` relation prefixes. Our new split, which we call ConceptNet-TN to denote True Negatives, consists of 36,210/3,278/3,278 train/validation/test triples. Again, the classes are balanced positive/negative, so accuracy is our main performance metric.

Note that because this dataset contains true (hard) negatives, we expect accuracy to be much lower than what we achieved in Table 4.2.

Baselines As baselines we consider several contrastive data augmentation approaches, all of which involve corrupting positive in-KB samples. We employ the following negative sampling baselines designed for **commonsense KBs**:

- **UNIFORM** [Li et al., 2016, Saito et al., 2018]: We replace the head phrase X_h or tail phrase X_t of each positive $(X_h, r, X_t)^+$ by uniformly sampling another phrase from the KB.
- **COMET** [Bosselut et al., 2019]: COMET is a version of GPT [Radford et al., 2018] that was fine-tuned to generate the tail phrase of a commonsense triple, conditioned on a head phrase and relation. To make COMET generate negatives, we prepend a "not" token to each positive head phrase X_h^+ and generate 10 tail phrases X_t^{COMET} for the modified head/relation prefix using beam search. Finally, we replace the tail phrase X_t in the positive with each X_t^{COMET} in turn, yielding negatives $(X_h^+, r, X_t^{\text{COMET}})$.

To investigate whether negative samplers tailored to encyclopedic knowledge can transfer to commonsense, we employ the following state-of-the-art baselines designed for **encyclopedic KBs**:

- **ROTATE-SA** [Sun et al., 2019]: For each positive instance, a pool of candidate negatives is generated with UNIFORM. The candidates are then scored by the (shallow, but state-of-the-art) RotatE KB embedding, and a negative is sampled from the candidate pool with probability proportional to the score distribution. We take the top 50% of self-adversarially generated statements as negative examples, in order of score descending, from the last epoch of training.

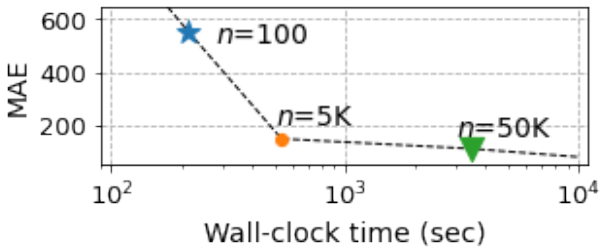


Figure 4.3: Lower left corner is best: Wall-clock time versus training loss (MAE) for NEGATER-∇ gradient magnitude prediction as training set size n increases.

- **SANS** [Ahrabian et al., 2020] is a graph-structural negative sampler that corrupts head/tail phrases of positive instances by sampling from the k-hop neighborhood of each KB entity. We set $k = 2$.

Finally, we **devise two additional baselines**:

- **SLOTS**: We replace the head phrase X_h (tail phrase X_t) of each positive $(X_h, r, X_t)^+$ by uniformly sampling from the set of phrases that appear in the head (tail) slot of KB triples mentioning relation r. We filter out all negative samples that appear in the KB.

- **ANTONYMS**: We tag each phrase in the KB as either a verb, noun, or adjective phrase using the SpaCy POS tagger.[1] Then, for each verb (noun, adjective) phrase, we replace the first verb (noun, adjective) token with a randomly selected antonym from either WordNet [Miller, 1998] or the gold lexical contrast dataset from [Nguyen et al., 2016b].

NegatER variants We generate out-of-KB candidates for NegatER with our k-NN approach using k=10, yielding around 570K candidates. We implement the NegatER candidate ranking methods as follows:

- **NEGATER-θ_r**: We rank candidates using fine-tuned BERT's classification scores. Since the scores are scaled differently by relation type, we combine the top-ranking 50% of candidates per relation and shuffle them.

- **NEGATER-∇**: We again use BERT to rank the candidates. To choose between the full-gradient and gradient-prediction approaches (Ch. 4.4.3.2), we train an MLP to predict gradient magnitudes and plot the mean absolute error training loss after 100 epochs for different training set sizes n. Figure 4.3 shows that even for n=5K examples, the loss quickly approaches zero. Therefore, for efficiency, we use an MLP trained on n=20K examples, which

[1] https://spacy.io/usage/linguistic-features#pos-tagging

takes around 1 hour to train and rank candidates on a single GPU, compared to an estimated 14 hours for the full-gradient approach. For a random sample of 100 candidates, the Pearson correlation coefficient between the true/predicted gradient magnitudes is ρ=0.982, indicating that the approximation is highly accurate.

- **No-ranking ablation**: Finally, in order to measure the importance of the LM ranking component of NegatER, we introduce an ablation which randomly shuffles the out-of-KB candidates rather than ranking them.

After we obtain each ranked list of candidates, we feed the statements as negative training examples to BERT/RoBERTa in order of rank descending.

4.6.2 Results and Discussion

For all performance metrics, we report averages over five trials to account for randomness in sampling and parameter initializations.

Accuracy comparison As shown in Table 4.3, training with the top-ranking negative examples from NegatER always yields the best accuracy for both LMs, up to 1.90% more than the baselines. Note that this improvement is achieved with changing how only *half* of the training examples (the negatives) are sampled. Notice also that our NegatER variants are the only samplers to offer **statistically significant improvements** over the UNIFORM baseline at $\alpha < 0.01$ for BERT and $\alpha < 0.05$ for RoBERTa (two-sided t-tests, five trials per model), signifying better-than-chance improvements.

Notice also that our most competitive baseline is SLOTS, which is a contrastive approach that samples new head/tail phrases from those appearing in the corresponding slots of the current relation r—that is, preserving the grammar (**R1**) of the relation. This confirms that grammatical negative samples are indeed more informative than nonsensical ones.

Encyclopedic versus commonsense? We hypothesize that the encyclopedic KB baselines ROTATE-SA and SANS underperform because such methods assume that the KB is a dense graph. While this is usually true for encyclopedic KBs, many commonsense KBs are highly sparse because entities are not disambiguated, which means that multiple phrases referring to the same concept may be treated as different entities in the KB. Malaviya et al. [2020] provide an illustrative example for comparison in their study of ranking-based commonsense KB completion: The popular encyclopedic KB completion benchmark FB15K-237 [Toutanova and Chen, 2015] is 75$\times$ denser than the ConceptNet benchmark studied in this chapter. Indeed, our SANS baseline assumes that there are plentiful entities within the k-hop neighborhood of a query entity, whereas in

Table 4.3: Accuracy on ConceptNet-TN using different negative sampling approaches: Our NegatER variants are the only negative samplers to offer statistically significant improvements over the popular UNIFORM baseline at $\alpha < 0.01$ ($^{\blacktriangle}$) for BERT and $\alpha < 0.05$ ($^{\triangle}$) for RoBERTa (two-sided t-test, five trials per model). **Bold/underline**: Best result per LM; Underline only: Second-best result per LM.

		BERT	RoBERTa
Baselines	UNIFORM	75.60 ± 0.24	75.55 ± 0.43
	COMET	76.04 ± 0.63	75.86 ± 0.75
	ROTATE-SA	75.30 ± 0.51	75.20 ± 0.37
	SANS	75.45 ± 0.38	75.17 ± 0.37
	SLOTS	$76.46 \pm 0.58^{\triangle}$	75.80 ± 0.25
	ANTONYMS	$76.06 \pm 0.30^{\triangle}$	75.58 ± 0.58
NegatER	θ_r ranking	$\mathbf{76.95 \pm 0.28}^{\blacktriangle}$	76.29 ± 0.59
	∇ ranking	$76.53 \pm 0.22^{\blacktriangle}$	$\mathbf{76.34 \pm 0.32}^{\triangle}$
	No ranking	75.61 ± 0.29	75.29 ± 0.19

reality there may be very few, and these entities may not be grammatical in context of the original positive (**R1**) nor thematically relevant (**R2**) to the query entity. Therefore, encyclopedic negative samplers may not be transferrable to commonsense KBs or other highly sparse KBs.

Ablation study Table 4.3 also indicates that the LM ranking component of NegatER is crucial for improving accuracy. Our no-ranking ablation leads to lower classification accuracy than both NEGATER-θ_r and NEGATER-∇. Empirically, we find that this is because the ranking step helps filter out false negatives generated by our k-NN candidate construction procedure.

Performance drill-down Finally, Table 4.4 provides precision and recall scores to further "drill down" into NegatER's effects. Evidently, the NegatER variants consistently yield the best precision, whereas there is no consistent winner in terms of recall. To understand why NegatER improves precision, we remind the reader that precision is calculated as $P = (TP)/(TP + FP)$, where TP stands for true positives and FP stands for false positives. Because training with NegatER examples helps the LMs better recognize hard negatives—examples that "look positive" but are really negative—the LM mislabels fewer negatives, decreasing the false positive rate.

4.7 Human Evaluation

Finally, we collect qualitative human judgments on the examples output by each negative sampler.

Table 4.4: NegatER consistently yields the highest precision on ConceptNet-TN among negative samplers because it lowers the false positive rate: Performance drill-down (stdevs omitted for space). $^{\blacktriangle}$, $^{\triangle}$: Significant improvement over UNIFORM at $\alpha < 0.01$ and $\alpha < 0.05$, respectively.

		BERT		RoBERTa	
		Prec.	Rec.	Prec.	Rec.
Baselines	UNIFORM	71.29	**85.83**	73.36	80.28
	COMET	73.73	80.99	73.47	81.02
	ROTATE-SA	74.83	76.59	73.70	78.48
	SANS	72.54	82.11	73.26	79.50
	SLOTS	75.21$^{\triangle}$	79.34	73.85	80.17
	ANTONYMS	72.55	83.98	72.98	**81.62**
NegatER	θ_r ranking	75.12$^{\blacktriangle}$	80.68	**75.92**$^{\blacktriangle}$	77.05
	∇ ranking	76.60$^{\blacktriangle}$	76.50	75.75$^{\blacktriangle}$	77.57
	No ranking	**76.81**$^{\blacktriangle}$	73.42	75.67$^{\triangle}$	74.78

4.7.1 Experimental Setup

Data To cover a diverse set of reasoning scenarios, we consider the `HasPrerequisite`, `HasProperty`, `HasSubevent`, `ReceivesAction`, and `UsedFor` relations from ConceptNet. For each relation and negative sampler, we take 30 negative statements at random, yielding 1,350 statements judged in total (5 relations $\times$ 9 negative samplers $\times$ 30 statements per method/relation).

Task We gather judgments for (**R1**) grammar on a binary scale (incorrect/correct) and (**R2**) thematic consistency of the head/tail phrases on a 4-point scale ("not consistent at all", "a little consistent", "somewhat consistent", "highly consistent"). To estimate the true negative rate, we also obtain truthfulness judgments on a 4-point scale ("not truthful at all", "sometimes true", "mostly true", "always true"). We recruit four annotators who are fluent in English. Among 50 statements shared across the annotators, we observe an average variance of 0.058 points on the 0/1 scale for **R1**, 0.418 points on the 4-point scale for **R2**, and 0.364 points on the 4-point truthfulness scale. According to previous work in commonsense KB construction [Romero et al., 2019], these values indicate high agreement. We provide the annotation instructions in Appendix A.

4.7.2 Results and Discussion

Table 4.5 compares normalized average judgment scores for **R1** and **R2**, as well as the percentage of statements labeled as "never true" (i.e., the true negative rate). The table suggests two takeaways. The first is that the requirements of negative knowledge are a tradeoff, as methods with higher true

Table 4.5: NegatER best trades off grammar (**R1**), consistency (**R2**), and the true negative rate, as measured by the percentage of statements labeled "never true": Human annotation scores, normalized out of 1. Relative and average ranks are provided because not all raw metrics are directly comparable—e.g., grammar (**R1**) is judged as binary, whereas consistency (**R2**) is graded.

		R1	**R2**	% "never true"	Avg rank
Baselines	UNIFORM	0.487 (9)	0.408 (7)	0.747 (3)	6.33 (9)
	COMET	0.580 (8)	**0.703** (1)	0.407 (9)	6.00 (8)
	ROTATE-SA	0.733 (7)	0.373 (8)	0.767 (2)	5.67 (7)
	SANS	0.760 (6)	0.532 (5)	0.633 (4)	5.00 (4)
	SLOTS	0.853 (5)	0.372 (9)	**0.773** (1)	5.00 (4)
	ANTONYMS	0.860 (4)	0.495 (6)	0.613 (5)	5.00 (4)
NegatER	θ_r ranking	0.880 (3)	0.635 (2)	0.413 (8)	4.33 (3)
	∇ ranking	**0.927** (1)	0.555 (4)	0.587 (6)	**3.67** (1)
	No ranking	0.920 (2)	0.592 (3)	0.560 (7)	4.00 (2)

Table 4.6: Our NEGATER-∇ variant best handles the tradeoff between consistency (**R2**) and truthfulness: Representative negative examples from the most competitive methods SLOTS, NEGATER-θ_r, and NEGATER-∇.

Method	Negative statement	Consistent?	True?
SLOTS	("open business", `HasPrerequisite`, "hide behind door")	A little	Never
	("go somewhere", `HasSubevent`, "bruise appears")	Not at all	Never
	("mailbox", `UsedFor`, "sleeping guests")	Not at all	Never
NEGATER-θ_r	("play baseball", `HasPrerequisite`, "join hockey team")	Somewhat	Never
	("comfort someone", `HasSubevent`, "talk with them")	Highly	Mostly
	("having a bath", `UsedFor`, "refreshing yourself")	Highly	Sometimes
NEGATER-∇	("hear news", `HasPrerequisite`, "record something")	A little	Never
	("drink water", `HasSubevent`, "inebriation")	Highly	Never
	("luggage trolley", `UsedFor`, "moving rocks")	Highly	Never

negative rates (ROTATE-SA, SLOTS) more often output ungrammatical or unconsistent statements, whereas methods that yield more consistent statements like COMET have a comparatively low true negative rate. The second is that **NEGATER-∇ best manages this tradeoff**, as it achieves the best average rank among all methods over the three criteria.

Finally, Table 4.6 provides examples of statements with consistency (**R2**) and truthfulness judgments. Again, it is evident that NEGATER-∇ best manages the tradeoffs of negative knowledge. In fact, it is the only negative sampler for which a majority of examples are rated both as "never true" (58.67%) and "somewhat consistent" or higher (62%).

4.8 Conclusion

In this chapter, we considered the problem of negative knowledge in commonsense KBs. Given the lack of research in this direction, we first rigorously defined negative commonsense knowledge. Next, we proposed a language model-based framework, NegatER, to address this problem. Importantly, NegatER does not require ground-truth negatives at any point, making it an effective choice when gold training examples are not available. We empirically demonstrated the strength of NegatER over many competitive baselines in multiple evaluations, including the strength of our fine-tuning approach, the task-based utility of NegatER statements, and the intrinsic quality of these statements. In particular, we observed the tradeoffs inherent to generating negative knowledge in terms of cohesiveness, downstream utility, and the true negative rate, and showed that NegatER uniquely strikes a balance between these competing criteria.

CHAPTER 5

Generating Novel Factual Knowledge

The material in this chapter is partially derived from the paper "CoDEx: A Comprehensive Knowledge Graph Completion Benchmark" [Safavi and Koutra, 2020], which appeared in the proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). It is also partially derived from ongoing work with the Allen Institute for Artificial Intelligence and the University of Washington, planned for submission in mid-2022 to a premier natural language processing conference.

5.1 Introduction

In this chapter, we continue to study the problem of machine knowledge representation with relational knowledge bases (KBs) and pretrained language models (LMs). However, whereas we focused on commonsense knowledge in the previous chapter, we now turn to encyclopedia-style knowledge about notable entities. In particular, we consider the task of automatic knowledge base completion, or KBC, which we introduced in Chapter 2.3.1. KBC is motivated by the observation that most large-scale KBs are high-precision but low-recall, missing many basic facts about notable people like their places of birth and occupation(s) [Galárraga et al., 2017, Weikum et al., 2021].

Our goal in this chapter is to introduce new resources and models for cross-modal KB link prediction—that is, KBC spanning structured relational *and* textual knowledge sources. Surprisingly, this direction has received relatively little attention in the literature compared to structure-only KBC, even though most KBs are linked to plentiful textual features like entity descriptions and Wikipedia infoboxes. Toward this goal, we identify two key gaps in the literature:

Lack of evaluation benchmarks As progress in artificial intelligence depends heavily on data, a relevant and high-quality benchmark is imperative to advancing the state of the art in cross-modal KBC. However, no comprehensive enyclopedic KBC benchmark combining structure and text exists. Currently, the prevailing approach for cross-modal KBC evaluation is to link subsets of

Table 5.1: Qualitative comparison of CoDEx to existing Freebase benchmarks.

	Freebase variants (FB15K, FB15K-237)	CoDEx datasets
Scope (domains)	Multi-domain, with a strong focus on awards, entertainment, and sports	Multi-domain, with focuses on writing, entertainment, music, politics, journalism, academics, and science
Scope (auxiliary data)	Various decentralized versions with, e.g., entity types [Xie et al., 2016], entity descriptions [Wang et al., 2021a], and randomly sampled negatives [Socher et al., 2013]	Centralized repository of three datasets with multilingual entity and relation descriptions, entity types, and manually verified hard negatives
Difficulty	FB15K has train/test leakage from inverse relations [Toutanova and Chen, 2015]; FB15K-237 has a high proportion of frequency-based relational patterns	Inverse relations removed from all datasets to avoid train/test leakage; few trivial patterns for the task of link prediction; manually annotated hard negatives for the task of triple classification

the deprecated Freebase KB [Bollacker et al., 2008] to textual sources like Wikipedia. However, we will show in this chapter that such datasets are limited in scope and difficulty, motivating the need for a new, purposefully designed KBC benchmark.

Lack of effective, efficient models KBC is most often formulated as ranking-based link prediction in a multi-relational graph [Ruffinelli et al., 2020]. Currently, the most effective KBC approaches in terms of ranking performance are efficient shallow KB embeddings that learn to rank potential links by modeling structural patterns in the KB [Sun et al., 2019, Ruffinelli et al., 2020]. However, these approaches do not leverage text. Conversely, pretrained language models (LMs) have recently shown promise in the link prediction task [Yao et al., 2019, Wang et al., 2021a]. However, they suffer from impractically slow inference time due to the combinatorial explosion problem of pairwise ranking with deep Transformer networks. It remains to be seen whether the efficiency and structural modeling ability of KB embeddings can be combined with the rich contextual representations from LMs for KBC.

5.1.1 Contributions

In this chapter, we propose to address both challenges. To address the need for cross-modal KBC evaluation benchmarks, we first present **CoDEx**, a set of knowledge graph **Co**mpletion **D**atasets **Ex**tracted from the Wikidata KB [Vrandečić and Krötzsch, 2014] and its sister project Wikipedia. Inasmuch as Wikidata is considered the successor of Freebase, CoDEx improves upon existing Freebase-based KBC benchmarks in terms of scope and difficulty (Table 5.1). We provide in-depth analysis of CoDEx to demonstrate its unique value as a KBC resource.

Next, we propose **CascadER**, a multi-stage **Cascad**e pipeline for **E**ntities and **R**elations. Cas-

cadER is an integrated approach to link prediction that takes full advantage of both structure and text in KBs. Motivated by the observation that structure-only KB embeddings are faster but potentially less expressive than deep contextual LMs for link prediction, CascadER uses LMs to *rerank* small sets of link prediction outputs from KB embeddings—exploiting their strengths on promising subsets of the problem space while avoiding their inefficiencies. Extensive experiments demonstrate that CascadER achieves remarkable and consistent gains of up to 9 points MRR (mean reciprocal rank) over strong structure-only baselines on challenging link prediction datasets, including but not limited to CoDEx.

Our contributions are as follows:

- **New comprehensive benchmark** (Ch. 5.3): To provide a starting point for future research in cross-modal KBC, we introduce CoDEx, a new KBC benchmark comprising structure and text. We provide extensive qualitative and quantitative analysis of CoDEx to demonstrate its unique merits compared to an existing KBC benchmark extracted from Freebase.
- **New model fusion approach** (Ch. 5.4): As a first step toward cross-modal ranking-based link prediction, we propose CascadER. CascadER is a multi-stage cascade ranking pipeline that uses expressive but computationally intensive LMs to rerank the partial outputs of shallow, fast KB embeddings. In order to control the "effectiveness-efficiency" tradeoff in multi-stage ranking [Wang et al., 2011], we propose a novel adaptive answer selection strategy that, for each link prediction query, predicts the number of top-ranked candidates to progress to the following tier.
- **Extensive experiments** (Ch. 5.5): We evaluate CascadER on five link prediction benchmarks including but not limited to CoDEx. CascadER achieves the highest MRR out of a suite of competitive baselines, up to 9 points improvement over the strongest structure-only baseline and 16 points improvement over the strongest text-only baseline. Moreover, CascadER exceeds the performance of our most competitive cross-modal ensembling baseline, while being more efficient by one or more orders of magnitude.

5.2 Related Work

In this section, we cover notable evaluation benchmarks and models for cross-modal KBC.

Cross-modal KBC benchmarks The most widely-used KBC benchmarks are extracted from Freebase [Bollacker et al., 2008]. FB15K was introduced by Bordes et al. [2013], It contains 14K entities, 1.3K relations, and 592K triples covering several domains, with a strong focus on awards,

entertainment, and sports. Toutanova and Chen [2015] introduced FB15K-237 to remedy data leakage in FB15K, which contains many test triples that can be predicted by inverting triples in the training set. FB15K-237 contains 14K entities, 237 relations, and 310K triples. Multiple studies have linked FB15K and FB15K-237 to Wikipedia to obtain aliases and textual descriptions of entities [Yao et al., 2019, Wang et al., 2021a]; we provide a detailed comparison of FB15K-237 to our proposed dataset CoDEx in Ch. 5.3.

Beyond Freebase, the NELL-995 dataset [Xiong et al., 2017] was taken from the Never Ending Language Learner (NELL) system [Mitchell et al., 2018], which continuously reads textual documents on the Web to obtain and update its knowledge. NELL-995, a subset of the 995th iteration of NELL, contains 75K entities, 200 relations, and 154K triples. While NELL-995 is general and covers many domains, its mean average precision was less than 50% around its 1000th iteration [Mitchell et al., 2018]. A higher-precision benchmark is YAGO3-10 [Dettmers et al., 2018], which is a subset of YAGO3 [Mahdisoltani et al., 2014] covering portions of Wikipedia, Wikidata, and WordNet. YAGO3-10 has 123K entities, 37 relations, and 1M triples mostly limited to facts about people and locations. While containing both structure and text, YAGO3-10 was shown to contain a high proportion of semantically duplicate relations [Akrami et al., 2020, Pezeshkpour et al., 2020].

To provide a high-level overview of benchmarks for this task, Table 5.2 considers a selection of papers published between 2014 and 2020 in the main proceedings of conferences where KBC embedding papers are most likely to appear: Artificial intelligence (AAAI, IJCAI), machine learning (ICML, ICLR, NeurIPS), and natural language processing (ACL, EMNLP, NAACL). The evaluation benchmarks covered in the table are FB15K [Bordes et al., 2013], WN18 [Bordes et al., 2013], FB15K-237 [Toutanova and Chen, 2015], WN18RR [Dettmers et al., 2018], FB13 [Socher et al., 2013], WN11 [Socher et al., 2013], NELL-995 [Xiong et al., 2017], YAGO3-10 [Dettmers et al., 2018], Countries [Bouchard et al., 2015], UMLS [McCray, 2003], Kinship [Kemp et al., 2006], Families [Hinton, 1986], and other versions of NELL [Mitchell et al., 2018].

Joint modeling on KBs The task of inferring novel links in KBs has been studied widely over the last decade. The most prevalent approaches are structure-only KB embeddings [Nickel et al., 2011, Bordes et al., 2013, Trouillon et al., 2016, Sun et al., 2019, Balazevic et al., 2019a, Ji et al., 2020]. That said, prior to the introduction of pretrained contextual language models, a few cross-modal structure and text modeling approaches were also proposed [Toutanova et al., 2015, 2016, Xie et al., 2016]. Such approaches rely on convolutional text representation architectures to obtain embeddings of entities or relations using, e.g., entity descriptions [Xie et al., 2016] or textual relation mentions [Toutanova et al., 2015]. These text-based embeddings are then composed using structural KB embedding scoring functions to rank potential novel links in the KB.

Table 5.2: A review of KBC benchmarks and evaluation tasks. Ranking refers to ranking-based link prediction, and classif. refers to classification-based link prediction.

		Datasets							Evaluation tasks		
	Reference	FB15K	FB15K-237	FB13	WN18	WN18RR	WN11	Other	Ranking	Classif.	Other
AAAI, IJCAI	[Wang et al., 2014]	✓		✓	✓		✓	FB5M	✓	✓	relation extraction (FB5M)
	[Lin et al., 2015b]	✓		✓	✓		✓	FB40K	✓	✓	relation extraction (FB40K)
	[Wang et al., 2015]							NELL (Location, Sports)	✓		
	[Nickel et al., 2016]	✓			✓			Countries	✓		
	[Lin et al., 2016]							FB24K	✓		
	[Wang et al., 2016]	✓			✓				✓		
	[Xiao et al., 2016a]	✓		✓	✓		✓		✓	✓	
	[Jia et al., 2016]	✓		✓	✓		✓		✓	✓	
	[Xie et al., 2016]	✓						FB15K-237+	✓	✓	
	[Shi and Weninger, 2017]	✓						SemMedDB, DBPedia	✓		fact checking (not on FB15K-237)
	[Dettmers et al., 2018]	✓	✓		✓	✓		YAGO3-10, Countries	✓		
	[Ebisu and Ichise, 2018]	✓			✓				✓		
	[Guo et al., 2018]	✓						YAGO37	✓		
	[Zhang et al., 2020]	✓	✓		✓	✓			✓		
	[Vashishth et al., 2020a]		✓			✓		YAGO3-10	✓		
ICML, ICLR, NeurIPS	[Yang et al., 2015]	✓			✓			FB15K-401	✓		rule extraction (FB15K-401)
	[Trouillon et al., 2016]	✓			✓				✓		
	[Liu et al., 2017]	✓			✓				✓		
	[Kazemi and Poole, 2018]	✓			✓				✓		
	[Das et al., 2018]		✓			✓		NELL-995, UMLS, Kinship, Countries, WikiMovies	✓		QA (WikiMovies)
	[Lacroix et al., 2018]	✓	✓		✓	✓		YAGO3-10	✓		
	[Guo et al., 2019]	✓	✓		✓			DBPedia-YAGO3, DBPedia-Wikidata	✓		entity alignment (DBPedia graphs)
	[Sun et al., 2019]	✓	✓		✓	✓			✓		
	[Zhang et al., 2019a]	✓	✓		✓	✓			✓		
	[Balazevic et al., 2019b]		✓			✓			✓		
	[Vashishth et al., 2020b]		✓			✓		MUTAG, AM, PTC	✓		graph classification (MUTAG, AM, PTC)
ACL, EMNLP, NAACL	[Ji et al., 2015]	✓		✓	✓		✓		✓	✓	
	[Guo et al., 2015]							NELL (Location, Sports, Freq)	✓	✓	
	[Guu et al., 2015]			✓			✓		✓	✓	
	[García-Durán et al., 2015]	✓						Families	✓		
	[Lin et al., 2015a]	✓						FB40K	✓		relation extraction (FB40K)
	[Xiao et al., 2016b]	✓		✓	✓		✓		✓	✓	
	[Nguyen et al., 2016a]	✓			✓				✓		
	[Xiong et al., 2017]		✓					NELL-995	✓		rule mining
	[Lin et al., 2018]		✓			✓		NELL-995, UMLS, Kinship	✓		
	[Nguyen et al., 2018]		✓			✓			✓		
	[Bansal et al., 2019]		✓			✓			✓		
	[Xu and Li, 2019]	✓	✓		✓	✓		YAGO3-10, Family	✓		
	[Balazevic et al., 2019a]	✓	✓		✓	✓			✓		
	[Nguyen et al., 2019]		✓			✓		SEARCH17	✓		personalized search (SEARCH17)
	[Nathani et al., 2019]		✓			✓		NELL-995, UMLS, Kinship	✓		
	[Jiang et al., 2019]	✓	✓		✓	✓			✓		

More recently, motivated by the wide applicability and processing power of Transformer language models, pretrained Transformer LMs like BERT [Devlin et al., 2019] have begun to gain traction for variants of the link prediction task [Yao et al., 2019, Kim et al., 2020, Daza et al., 2021, Wang et al., 2021a, Nadkarni et al., 2021]. The approaches most related to CascadER are the ensembles considered by Wang et al. [2021a] and Nadkarni et al. [2021], both of which construct additive ensembles of structural KB embeddings and contextual LMs. However, neither study considers cascaded multi-stage ranking, which, as we will show subsequently, is key to effective and efficient cross-modal link prediction.

Cascade models Multi-stage cascade ensembles have been successful in computer vision [Viola and Jones, 2001, Wang et al., 2022] and text retrieval [Wang et al., 2011, Chen et al., 2017, Gallagher et al., 2019]. Recently, several studies have proposed to use BERT as a late-stage ranker in multi-stage document retrieval [Nogueira et al., 2019] and passage retrieval [Matsubara et al., 2020] pipelines. Similar to our work, these studies are motivated by the observation that using BERT in a multi-stage cascaded setting can significantly boost retrieval accuracy while maintaining efficiency [Lin et al., 2021]. Yet other studies have attempted to balance the effectiveness-efficiency tradeoff by proposing dual-encoding architectures that are much more efficient (but usually less effective) than single-encoder BERT models for information retrieval [Reimers and Gurevych, 2019, Humeau et al., 2020, Karpukhin et al., 2020, Xiong et al., 2020a, Khattab and Zaharia, 2020]. Our work builds upon all of these important insights, which have been instrumental in scaling contextual LMs to large-scale ranking tasks in information retrieval and natural language processing. As far as we are aware, we are the first to bridge these ideas with the traditional graph learning task of link prediction.

5.3 Dataset Construction

We have already established in Ch. 5.2 that no suitable benchmark for KBC across structure and text exists. Motivated by this gap in the literature, we define the following desiderata for a new KBC benchmark:

R1 Covers diverse structural patterns: It has been shown that real-world encyclopedic KBs tend to bias toward a few "popular" entities [Meij et al., 2020]. However, entity degrees in many KBs also exhibit a very long tail [Li et al., 2017]. An ideal KBC benchmark should capture structural variation, and should contain both high-frequency and rare entities.

R2 Covers diverse textual content: To enable text-augmented KBC, new KBC benchmarks should link all entities and relations to diverse textual contexts. By diverse we mean that

linked texts should be available at multiple levels of granularity (e.g., entity aliases versus full entity descriptions) and multiple languages.

R3 **Appropriately difficult**: Several previous studies have analyzed existing KBC benchmarks and shown that they contain train/test leakage [Toutanova and Chen, 2015, Akrami et al., 2020, Pezeshkpour et al., 2020]. A new benchmark should avoid such leakage where possible, and should not be easily solved with trivial non-learning baselines.

In the remainder of this section, we introduce CoDEx, describe how we collected its various components, and provide qualitative and quantitative analyses to confirm that it fulfills **R1**, **R2**, and **R3**.

5.3.1 Structural Data Collection

Seeds We began our data collection by querying Wikidata [Vrandečić and Krötzsch, 2014], which organizes an overlapping subset of information from Wikipedia in relational KB form. First, we collected an initial set of triples using snowball sampling [Goodman, 1961]. To collect seeds, we manually defined a broad seed set of entity and relation types common to 13 domains: Business, geography, literature, media and entertainment, medicine, music, news, politics, religion, science, sports, travel, and visual art. We then queried Wikidata for statements of the form (*head entity of seed type*, *seed relation type*, ?), and retrieved an initial set of 380K entities, 75 relations, and 1.1M triples. Table 5.3 provides all seed entity and relation types used to collect CoDEx. Each type is given first by its natural language label and then by its Wikidata unique ID: Entity IDs begin with Q, whereas relation (property) IDs begin with P. For the entity types that apply to *people* (e.g., actor, musician, journalist), we retrieved seed entities by querying Wikidata using the *occupation* relation. For the entity types that apply to *things* (e.g., airline, disease, tourist attraction), we retrieved seed entities by querying Wikidata using the *instance of* and *subclass of* relations.

Data filtering To meet requirement **R1** of diverse structural patterns, we filtered the initial triples to k-cores, which are maximal subgraphs $\mathcal{G}'$ of a given graph $\mathcal{G}$ such that every node in $\mathcal{G}'$ has a degree of at least k [Batagelj and Zaveršnik, 2011]. Using this approach we constructed three CoDEx datasets (Table 5.4): **CoDEx-S** (k = 15), **CoDEx-M** (k = 10), and **CoDEx-L** (k = 5). The datasets have 32K, 185K, and 551K triples, with densities on the order of magnitude of 10^{-2}, 10^{-3}, and 10^{-4}, respectively.

Note that because each CoDEx KB has a different level of sparsity, we can expect different types of method to be more competitive depending on the dataset. In particular, structure-only KBC approaches should be more competitive on CoDEx-S because each entity participates in at least 15 different relationships. By contrast, on CoDEx-M and CoDEx-L, we expect textual

Table 5.3: The entity and relation types (Wikidata IDs in parentheses) we manually defined to seed our data collection. For the entity types that apply to *people* (e.g., actor, musician, journalist), we retrieved seed entities by querying Wikidata using the *occupation* relation. For the entity types that apply to *things* (e.g., airline, disease, tourist attraction), we retrieved seed entities by querying Wikidata using the *instance of* and *subclass of* relations.

	Seed types
Entities	actor (Q33999), airline (Q46970), airport (Q1248784), athlete (Q2066131), book (Q571), businessperson (Q43845), city (Q515), company (Q783794), country (Q6256), disease (Q12136), engineer (Q81096), film (Q11424), government agency (Q327333), journalist (Q1930187), lake (Q23397), monarch (Q116), mountain (Q8502), musical group (Q215380), musician (Q639669), newspaper (Q11032), ocean (Q9430), politician (Q82955), record label (Q18127), religion (Q9174), religious leader (Q15995642), religious text (Q179461), scientist (Q901), sports league (Q623109), sports team (Q12973014), stadium (Q483110), television program (Q15416), tourist attraction (Q570116), visual artist (Q3391743), visual artwork (Q4502142), writer (Q36180)
Relations	airline alliance (P114), airline hub (P113), architect (P84), architectural style (P149), author (P50), capital (P36), cast member (P161), cause of death (P509), chairperson (P488), chief executive officer (P169), child (P40), continent (P30), country (P17), country of citizenship (P27), country of origin (P495), creator (P170), diplomatic relation (P530), director (P57), drug used for treatment (P2176), educated at (P69), employer (P108), ethnic group (P172), field of work (P101), foundational text (P457), founded by (P112), genre (P136), head of government (P6), head of state (P35), headquarters location (P159), health specialty (P1995), indigenous to (P2341), industry (P452), influenced by (P737), instance of (P31), instrument (P1303), language of work or name (P407), languages spoken, written, or signed (P1412), legal form (P1454), legislative body (P194), located in the administrative terroritorial entity (P131), location of formation (P740), medical condition (P1050), medical examinations (P923), member of (P463), member of political party (P102), member of sports team (P54), mountain range (P4552), movement (P135), named after (P138), narrative location (P840), notable works (P800), occupant (P466), occupation (P106), official language (P37), parent organization (P749), part of (P361), place of birth (P19), place of burial (P119), place of death (P20), practiced by (P3095), product or material produced (P1056), publisher (P123), record label (P264), regulated by (P3719), religion (P140), residence (P551), shares border with (P47), sibling (P3373), sport (P641), spouse (P26), studies (P2578), subclass of (P279), symptoms (P780), time period (P2348), tributary (P974), unmarried partner (P451), use (P366), uses (P2283)

Table 5.4: Statistics of CoDEx. We compute density as the number of edges in the KB across train, dev, and test, divided by the maximal number of undirected edges $(|\mathcal{V}| \cdot |\mathcal{V} - 1|)/2$.

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	# train	# dev	# test	Density
CoDEx-S	2,034	42	32,888	1827	1828	0.01767
CoDEx-M	17,050	51	185,584	10,310	10,311	0.00142
CoDEx-L	77,951	69	551,193	30,622	30,622	0.00020

Table 5.5: Average number of words for entities in each CoDEx dataset. Note that the larger datasets have shorter Wikipedia extracts on average because these datasets have a larger proportion of entities with either very short Wikipedia pages or no Wikipedia page at all. All text lengths are reported for English.

	Wikidata aliases	Wikidata descriptions	Wikipedia extracts
CoDEx-S	2.05	5.55	259.24
CoDEx-M	2.21	4.60	159.48
CoDEx-L	2.24	3.64	96.01

Table 5.6: Multilingual coverage in CoDEx. We compute multilingual coverage over all labels, descriptions, and Wikipedia extracts successfully retrieved for the respective dataset in Arabic (ar), German (de), English (en), Spanish (es), Russian (ru), and Chinese (zh).

	ar	de	en	es	ru	zh
CoDEx-S	77.38	91.87	96.38	91.55	89.17	79.36
CoDEx-M	75.80	95.20	96.95	87.91	81.88	69.63
CoDEx-L	67.47	90.84	92.40	81.30	71.12	61.06

KBC methods to fill in more "gaps" when structural information is not available. We verify these hypotheses experimentally in Ch. 5.5.

Train/test splitting To minimize train/test leakage as per requirement **R3**, we removed inverse relations from each dataset following Toutanova and Chen [2015]. We computed (*head*, *tail*) and (*tail*, *head*) overlap between all pairs of relations, and removed each relation whose entity pair set overlapped with that of another relation more than 50% of the time. Finally, we split each dataset into 90/5/5 train/validation/test triples assuming a transductive setting, in which all entities appearing in the test set also appear at least once in training.

5.3.2 Textual Data Collection

We next collected the textual component of CoDEx. To ensure diversity of texts, following requirement **R2**, we gathered textual contexts from multiple sources, and across multiple languages.

Textual sources We gathered **Wikidata aliases and descriptions** for all entities and relations, alongside **Wikipedia page extracts**, introduction section only, for all entities. As shown in Table 5.5, the English Wikidata aliases and descriptions are relatively short, comprising around 2 words per alias and 3 to 5 words per description. By contrast, the English Wikipedia extracts are relatively long, comprising around 100 or more words on average.

Languages considered We collected all textual information where available in Arabic, German, English, Spanish, Russian, and Chinese. We chose these languages because they are all relatively well-represented on Wikidata [Kaffee et al., 2017]. Table 5.6 provides the coverage by language for each CoDEx dataset.

5.3.3 Structure Analysis

In this section, we analyze the structure of CoDEx to verify that we have met requirements **R1** and **R3**. Toward **R1**, we analyze various types of logical relation patterns in each CoDEx dataset. Toward **R3**, we conduct a comparative case study between CoDEx-M and FB15K-237, an existing and widely-used link prediction benchmark of comparable size and content. We show that while the two datasets have similar structure on a shallow level, CoDEx is a more challenging link prediction benchmark because it contains fewer trivial frequency patterns. This difficulty of CoDEx as a structure-only benchmark also implies that the textual content in CoDEx may be more useful for the prediction task than for FB15K-237; we will show empirically in Chapter 5.5 that this is true.

5.3.3.1 Logical Relation Patterns

To elucidate the types of relational reasoning necessary for models to perform well on CoDEx, we analyze the presence of learnable binary relation patterns within CoDEx. The three main types of such patterns in knowledge bases are **symmetry**, **inversion**, and **compositionality** [Trouillon et al., 2019, Sun et al., 2019]. We address symmetry and compositionality here, and omit inversion because we specifically removed inverse relations to avoid train/test leakage.

Symmetry Symmetric relations are relations r for which $(h, r, t) \in \mathcal{G}$ implies $(t, r, h) \in \mathcal{G}$. For each relation, we compute the number of its (*head*, *tail*) pairs that overlap with its (*tail*, *head*)

Table 5.7: Relational patterns in CoDEx. For symmetry, we give the proportion of triples containing a symmetric relation. For composition, we give the proportion of triples participating in a rule of length two or three.

	CoDEx-S	CoDEx-M	CoDEx-L
Symmetry	17.46%	4.01%	3.29%
Composition	10.09%	16.55%	31.84%

pairs, divided by the total number of pairs, and take those with 50% overlap or higher as symmetric. CoDEx datasets have five such relations: *diplomatic relation*, *shares border with*, *sibling*, *spouse*, and *unmarried partner*. Table 5.7 gives the proportion of triples containing symmetric relations per dataset. Symmetric patterns are more prevalent in CoDEx-S, whereas the larger datasets are mostly **antisymmetric**, i.e., $(h, r, t) \in \mathcal{G}$ implies $(t, r, h) \notin \mathcal{G}$.

Composition Compositionality captures **path rules** of the form $(h, r_1, x_1), \ldots, (x_n, r_n, t) \rightarrow (h, r, t)$. To identify compositional paths, we use the AMIE3 system [Lajus et al., 2020], which outputs rules with confidence scores that capture how many times those rules are seen versus violated, to identify paths of lengths two and three; we omit longer paths as they are relatively costly to compute. We identify 26, 44, and 93 rules in CoDEx-S, CoDEx-M, and CoDEx-L, respectively, with average confidence (out of 1) of 0.630, 0.556, and 0.459.

Table 5.7 gives the percentage of triples per dataset participating in a discovered rule. Evidently, composition is especially prevalent in CoDEx-L. An example rule in CoDEx-L is "if *X* was founded by *Y*, and *Y*'s country of citizenship is *Z*, then *X*'s country of origin is *Z*" (confidence 0.709).

5.3.3.2 Difficulty Comparison

Next, to analyze CoDEx from a "difficulty" perspective, we conduct a brief comparison between CoDEx-M and FB15K-237, a similarly-sized encyclopedic knowledge benchmark for link prediction. We show that CoDEx is a more challenging link prediction benchmark because fewer of its test queries can be solved by a simple frequency baseline, meeting requirement **R3** for a new KBC benchmark.

Baseline We devise a simple link prediction baseline. Let $(h, r, ?)$ be a test query. Our baseline scores candidate tail entities by their relative frequency in the tail slot of all training triples mentioning r, filtering out tail entities t for which (h, r, t) is already observed in the training set. If all tail entities t are filtered out, we score entities by frequency before filtering. The logic of our

Table 5.8: Overall performance (MRR) of our frequency baseline versus the best structure-only KGE model per benchmark. "Improvement" refers to the improvement of the KGE over the baseline.

	Baseline	Embedding	Improvement
FB15K-237	0.236	0.356	+0.120
CoDEx-M	0.135	0.337	+0.202

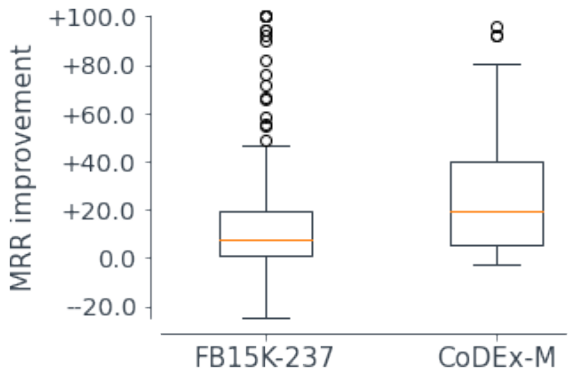


Figure 5.1: Improvement in MRR of the embedding over our baseline per relation type. Negative means that our baseline outperforms the embedding.

approach works in reverse for $(?, r, t)$ queries. In evaluating our baseline, we follow LibKGE's protocol for breaking ties in ranking (i.e., for entities that appear with equal frequency) by taking the mean rank of all entities with the same score.

Task and metrics Since our baseline only uses entity and relation frequency information and does not leverage text, we compare our baseline to a highly competitive structure-only KGE on each dataset: RESCAL for FB15K-237 [Ruffinelli et al., 2020] and ComplEx for CoDEx-M. We evaluate performance with MRR and Hits@10. Beyond overall performance, we also compute *per-relation improvement* of the respective embedding over our baseline in percentage points MRR and Hits@10. This measures the benefit of learning each relation over using a simple frequency rule.

Results and discussion Table 5.8 compares the overall performance of our baseline versus the best embedding per dataset, and Figure 5.1 shows the improvement of the respective embedding over our baseline per relation type on each dataset.

Evidently, the improvement of the embedding is much smaller on FB15K-237 than CoDEx-M. In fact, our baseline performs on par with or even outperforms the embedding on FB15K-237

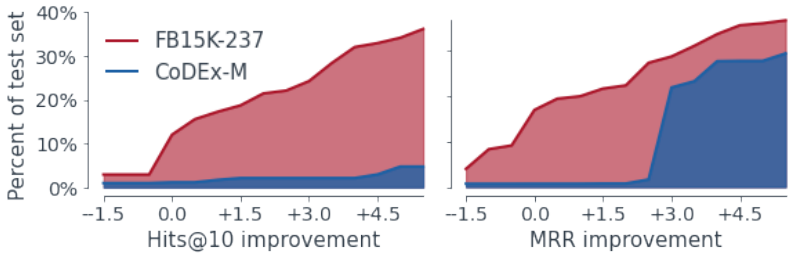


Figure 5.2: Empirical cumulative distribution function of embedding improvement over the baseline.

for some relation types. To further explore these cases, Figure 5.2 gives the empirical cumulative distribution function of improvement, which shows the percentage of test triples for which the level of improvement is less than or equal to a given value on each dataset. Surprisingly, the improvement is less than five percentage points for nearly 40% of FB15K-237's test set, and is zero or negative 15% of the time. By contrast, our baseline is significantly weaker than the embedding on CoDEx-M.

The disparity in improvement is because FB15K-237 has more relations that are highly skewed toward a few entities. For example, our baseline achieves perfect performance over all $(h, r, ?)$ queries for the `/common/topic/webpage./common/webpage/category` relation because this relation has only one unique tail entity. In total, 11 relations in FB15K-237 have one unique tail entity, and these relations account for 3.22% of all tail queries in FB15K-237; 15.98% of test triples in FB15K-237 contain relations that are skewed 50% or more toward a single head or tail entity, whereas only 1.26% of test triples in CoDEx-M contain skewed relations of this type. Furthermore, around 12.7% of test queries in FB15K-237 contain relation types that connect entities to small fixed sets of literal values. For example, each head entity that participates in the FB15K-237 relation `/travel/travel\_destination/climate./travel/travel\_destination\_monthly\_climate/month` is connected to the same 12 tail entities (months) throughout train, validation, and test. This makes prediction trivial with our baseline: By filtering out the tail entities already seen in train, only a few (or even one) candidate tail(s) are left in test, and the answer is guaranteed to be within these candidates.

We conclude that while FB15K-237 is a valuable dataset, CoDEx is more appropriately difficult for link prediction. Additionally, we note that in FB15K-237, all validation and test triples containing entity pairs directly linked in the training set were deleted [Toutanova and Chen, 2015], meaning that symmetry cannot be tested for in FB15K-237. Given that CoDEx datasets contain both symmetry and compositionality, CoDEx is more suitable for assessing how well models can *learn* relation patterns that go beyond frequency.

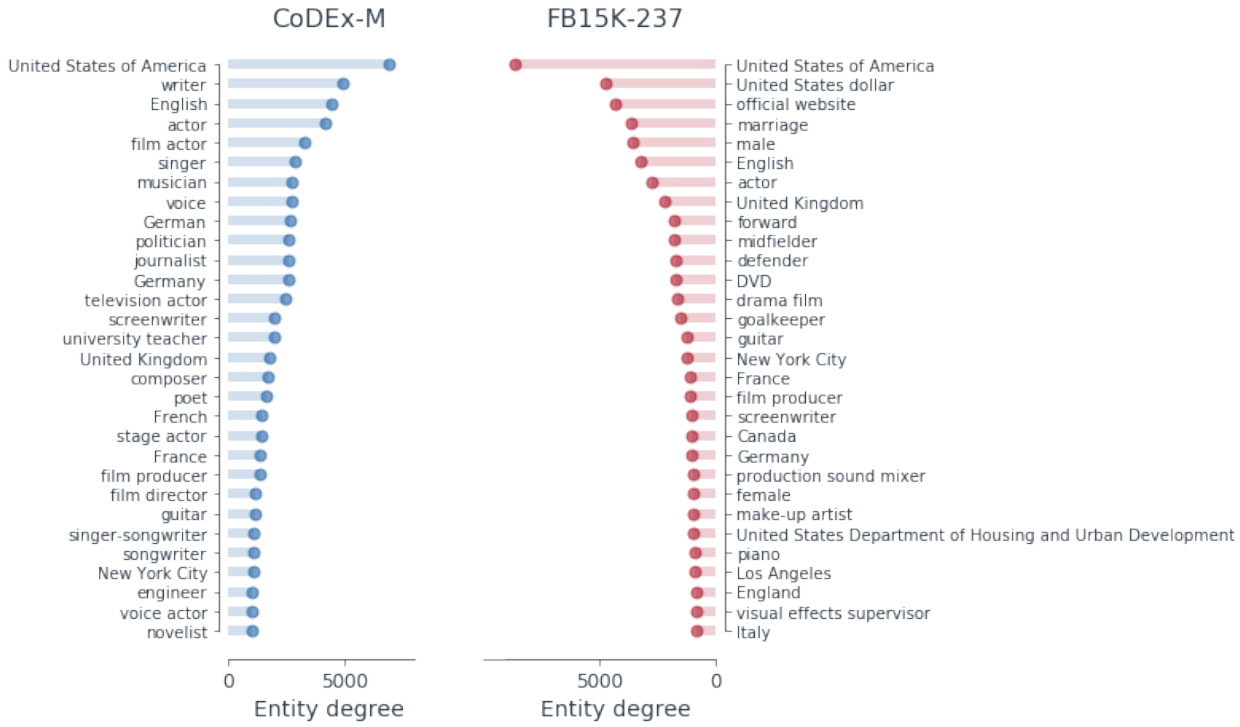


Figure 5.3: Top-30 most frequent entities in CoDEx-M and FB15K-237.

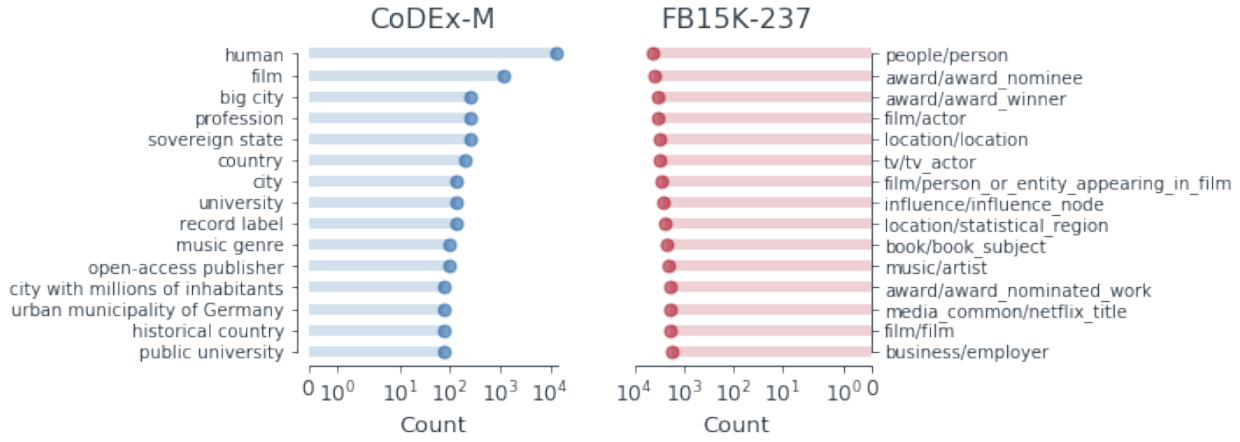


Figure 5.4: Top-15 most frequent entity types in CoDEx-M and FB15K-237.

5.3.4 Content Analysis

Finally, we briefly analyze the content in CoDEx-M, again by comparing it to FB15K-237, to confirm that it meets our stated goal of textual diversity (**R2**).

Entities As shown in Figure 5.3 and Figure 5.4, both CoDEx-M and FB15K-237 are biased toward developed Western countries and cultures. However, CoDEx-M is more diverse in domain.

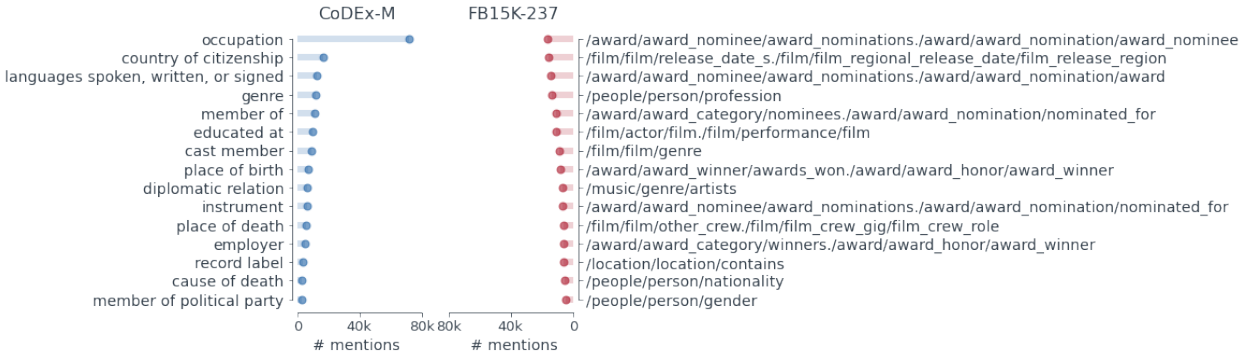


Figure 5.5: Top-15 most frequent relations in CoDEx-M and FB15K-237.

It covers academia, entertainment, journalism, politics, science, and writing, whereas FB15K-237 covers mostly entertaiment and sports. FB15K-237 is also much more biased toward the United States in particular, as five of its top-30 entities are specific to the US: *United States of America*, *United States dollar*, *New York City*, *Los Angeles*, and *United States Department of Housing and Urban Development*.

Relations Figure 5.5 compares the top-15 relations by mention count in the two datasets. The most common relation in the former is *occupation*, which is because most people on Wikidata have multiple occupations listed. By contrast, the frequent relations in FB15K-237 are mostly related to awards. In fact, over 25% of all triples in FB15K-237 belong to the */award* domain.

It is also worth noting that the Freebase-style relations are arguably less interpretable than those in Wikidata. Whereas Wikidata relations have concise natural language labels, the Freebase relation labels are hierarchical, often at five or six levels of hierarchy (Figure 5.5). Moreover, all relations in Wikidata are binary, whereas some Freebase relations are n-nary. Specifically, Freebase used a special type of entity called Compound Value Type (CVT) to express *n-ary* relationships consisting of multiple fields like literal values, temporal occurrences, etc [Tanon et al., 2016]. Many relation types in FB15K-237 were created by traversing through CVTs to yield compound binary relation types, which are arguably difficult to reason about.

5.4 Methodology

Having introduced our new dataset CoDEx for cross-modal KBC and thoroughly established its merits as a link prediction benchmark, we now consider the important next step: Effective and efficient modeling for link prediction over structure and text. As introduced in Chapter 2.2.1, we consider the task of knowledge base completion (KBC) formulated as ranking-based link predic-

tion in a multi-relational graph $\mathcal{G}$ consisting of entities $\mathcal{V}$, relations $\mathcal{R}$, and factual (*head*, *relation*, *tail*) triples $(h, r, t) \in \mathcal{V} \times \mathcal{R} \times \mathcal{V}$. Recall that the link prediction task consists of two settings. In the first, given a tail query $(h, r, ?)$, score all entities $\hat{t} \in \mathcal{V}$ by their likelihood that they answer the query such that the true tail entity t is ranked as high as possible. In the second, given a head query $(?, r, t)$, score all entities $\hat{h} \in \mathcal{V}$ by the likelihood that they answer the query.

In this section, we first cover the key differences in structure- versus text-based approaches to link prediction, and subsequently introduce CascadER, a novel cross-modal sequential reranking architecture, to bridge the gap.

5.4.1 Existing Approaches

We define a link prediction model as a scoring function $f : \mathcal{V} \times \mathcal{R} \times \mathcal{V} \to \mathbb{R}$ that takes as input a triple from $\mathcal{G}$ and outputs a real value indicating the plausibility of that triple. At inference time, assume that we have N_{test} link prediction queries of interest, half of type $(h, r, ?)$ and half of type $(?, r, t)$. For each query we have $|\mathcal{V}|$ potential answers, which are the entities in the KB. We define a link prediction query-answer score matrix $\mathbf{S} \in \mathbb{R}^{N_{\text{test}} \times |\mathcal{V}|}$, in which S_{ij} denotes the predicted probability that entity j answers link prediction query i.

5.4.1.1 Single-Modality Approaches

Structure-based models The most competitive structure-based approaches to link prediction are shallow knowledge graph embeddings (**KGE**s), which are decoder models that train entity and relation embeddings to directly optimize for the link prediction ranking task. For more details on KGEs, we refer the reader to Chapter 2.4.1.

Text-based models As discussed in our review in Ch. 3.4, contextual language models may be adapted to various knowledge representation tasks, including knowledge base construction and completion. Assuming that the input KB is linked to text, let $X_h = [w_1, \ldots, w_h]$ denote the tokens comprising the description of head entity h. Likewise, let $X_t = [w_1, \ldots, w_t]$ the description of tail entity t. **Cross-encoding** language model approaches for link prediction feed each full triple sequence to a single encoder as [`[CLS]`, X_h, `[SEP]`, X_t, `[SEP]`], where `[CLS]` and `[SEP]` refer to the LM's special classification and delimiter tokens. At the output of the encoder, these approaches stack a binary triple classification layer [Yao et al., 2019], optionally with other layers for related KB completion tasks like relation disambiguation [Kim et al., 2020].

By contrast, **dual-encoding** approaches assume two encoder language models instead of one, potentially with shared weights [Reimers and Gurevych, 2019, Humeau et al., 2020, Wang et al., 2021a]. One encoder takes as input the head tokens [`[CLS]`, X_h, `[SEP]`], and the other takes

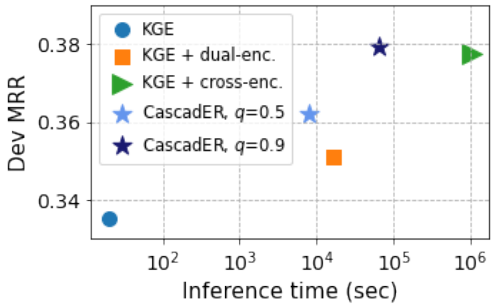


Figure 5.6: CascadER maintains effectiveness (validation MRR) while improving efficiency (inference wall-clock time) by one or more orders of magnitude compared to our most competitive ensemble baseline on CoDEx-M. Dual-enc. refers to a dual-encoder LM, and cross-enc. refers to a cross-encoder LM; we discuss the differences in these architectures in § 5.4.1. For CascadER, we consider a three-tier structure with dynamic answer pruning at quantiles $q = 0.5$ and $q = 0.9$ (§ 5.4.4).

as input the tail tokens `[[CLS]`, X_t, `[SEP]]`. Both encoders output embeddings of their input sequences, and these output embeddings are trained to maximize similarity for observed (*head*, *tail*) pairs in the KB, again optionally with an additional relation disambiguation loss.

5.4.1.2 Cross-Modal Approaches

Recently, it has been proposed to integrate structure and text into link prediction by ensembling KGEs and LMs with additive reweighting [Wang et al., 2021a, Nadkarni et al., 2021]. Given a query i and candidate answer j, additive reweighting outputs a new link prediction score as the convex combination of the base models' scores:

$$S_{ij}^{\text{ens}} = \alpha \cdot S_{ij}^{\text{KGE}} + (1 - \alpha) \cdot S_{ij}^{\text{LM}},$$

where the weight $\alpha \in [0, 1]$ is a hyperparameter tuned on a held-out set.

As shown in Figure 5.6, additive ensembling can significantly improve link prediction performance, especially with cross-encoder LMs. However, ensembling comes at the expense of computational efficiency. Figure 5.6 also shows that ensembling with an LM increases inference complexity over the KGE by several orders of magnitude. This added expense is due to several factors. First, pretrained Transformer LMs encode entity descriptions using multiple layers (typically 12 or more) prior to scoring, and the complexity of encoding scales quadratically with the input length. Second, while dual-encoders can pre-compute the embeddings of all entity descriptions [Karpukhin et al., 2020] and score new links via relatively fast dot products between entity embedding pairs, cross-encoders must jointly encode and score each query/answer pair separately,

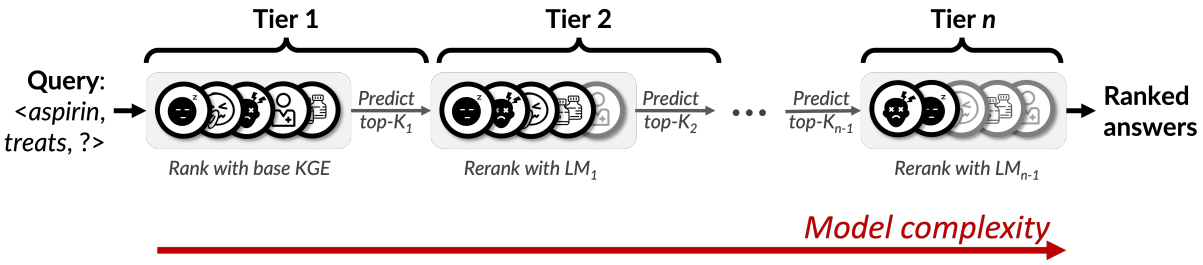


Figure 5.7: CascadER sequential reranking architecture.

which makes them impractically slow for large-scale text ranking [Reimers and Gurevych, 2019].

5.4.2 CascadER Overview

Assuming that we are willing to pay some computational cost to improve link prediction performance, how can we achieve the *effectiveness* of the cross-encoder ensemble while maintaining *efficiency* closer to that of the dual-encoder ensemble, as shown in Figure 5.6? Our answer is **CascadER**, a cross-modal progressive refinement architecture that combines the best of both worlds. As illustrated in Figure 5.7, CascadERis a tiered architecture that treats one or more LMs as rerankers to a base KGE. The key idea of CascadER is that we can *selectively* invoke LMs to reweight the base KGE's most promising candidate predictions, which cuts down on the computational cost of language modeing while still benefiting from the performance gains of ensembling.

More formally, assume we have a set of $n \geq 2$ trained link prediction models $\{f^{(i)}, i = 1 \dots n\}$ consisting of one KGE and one or more LMs. We sort the models by computational complexity, leading to an ordered sequence $(f^{(1)}, \dots, f^{(n)})$ in which $f^{(1)}$ is the KGE and the subsequent models are LMs in ascending order of complexity (i.e., dual-encoders before cross-encoders). We use the KGE to score all query/answer pairs in the inference set, leading to a score matrix $\mathbf{S} \in \mathbb{R}^{N_{\text{test}} \times |\mathcal{V}|}$ in which S_{ij} denotes the KGE probability of entity j answering the i-th link prediction query. Then, at each tier $t = 1, \dots, n-1$, we apply a pruning function that, for each query i, selects a subset of candidate answer entities $\mathcal{V}_i^{(t)} \subseteq \mathcal{V}$ to be reranked by the next-tier LM $f^{(t+1)}$; we postpone the discussion of pruning strategies to Chapters 5.4.3 and 5.4.4.

For query i and candidate answer j, we define the additive reranking function between tiers t and $t+1$ as follows:

$$\mathbb{I}_{j \in \mathcal{V}_i^{(t)}} \left[\alpha^{(t)} \cdot S_{ij}^{(t)} + \left(1 - \alpha^{(t)}\right) \cdot S_{ij}^{(t+1)} \right] + \mathbb{I}_{j \notin \mathcal{V}_i^{(t)}} \left[S_{ij}^{(t)} \right],$$

where $\mathbb{I}$ denotes the set indicator function, $S_{ij}^{(t)}$ denotes the query-answer plausibility score output by the model at tier t, and $\alpha^{(t)} \in [0, 1]$ is a hyperparameter that controls the additive influence of

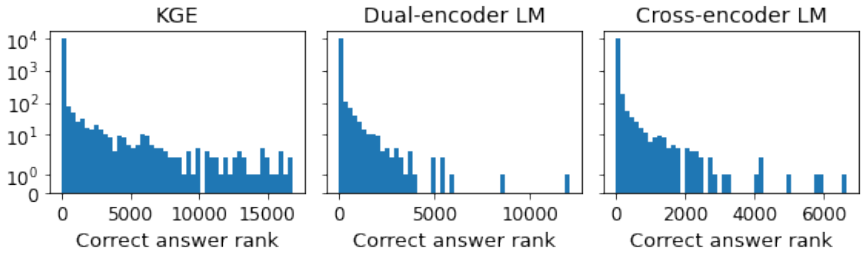


Figure 5.8: Ranks of the gold answer entities on the validation set of CoDEx-M.

model $f^{(t)}$ in reranking the candidates in $\mathcal{V}_i^{(t)}$.

5.4.3 Static Candidate Pruning

Following the effectiveness-efficiency tradeoff, we have two goals for candidate pruning with CascadER: **(1)** For each link prediction query $(h, r, ?)$ or $(?, r, t)$, progress as many candidate entities to the following tier as possible to ensure coverage of all promising candidates; and **(2)** Progress as few candidates as possible to maintain efficiency and avoid invoking the next-tier rerankers on less-promising candidates.

A straightforward pruning approach used in information retrieval to balance these two goals is to progress only the top-k candidates from tier to tier given a global value of k [Wang et al., 2011, Matsubara et al., 2020]. Formally, given query i and a selected value of k, we define static pruning as selecting the subset of candidates $\mathcal{V}_i^{(t)}$ such that

$$\mathcal{V}_i^{(t)} = \underset{\mathcal{V}_i^{(t)} \subset \mathcal{V} \text{ and } |\mathcal{V}_i^{(t)}|=K}{\arg\max} \sum_{j=1}^{|\mathcal{V}|} S_{ij}^{(t)}.$$

Of course, the challenge with this pruning approach is selecting the correct value of k. One solution is to set k ad-hoc, e.g., $k = 100$ or $k = 1000$ [Matsubara et al., 2020]. However, setting the wrong value of k may result in suboptimal performance from the effectiveness or efficiency perspectives.

To address this challenge, we propose a more principled strategy that searches for the best value of k per dataset in terms of MRR on a held-out set. Given tier t and hold-out query i, we obtain the cascade's rank $R_i^{(t)}$ of the gold answer entity. We construct a distribution of ranks $R_i^{(t)}$ over all hold-out queries, and use quantiles of this distribution to choose the grid of k over which to search. For example, quantiles of 0.5, 0.75, and 0.9 means that we search over k equal to the median, 75th percentile, and 90th percentile of ranks $R_i^{(t)}$, which helps ensure that our choice of k is data-driven. We use quantiles because the distributions of gold answer ranks are non-normally distributed, as shown in Figure 5.8.

5.4.4 Dynamic Candidate Pruning

We propose to extend our static pruning strategy to handle more nuanced differences between queries. The motivation for this approach is that, depending on the structural and textual information available in the dataset, some queries may be more difficult than others. We hypothesize that if we can assess the difficulty of each query, we can more accurately determine the amount of reranking necessary for each query at each tier of the cascade, in order to better balance effectiveness and efficiency.

We formulate this selection strategy as a *dynamic* pruning approach in which, for tier t and query i, we predict an integer $\hat{k}_i^{(t)}$ that represents the number of candidates to pass to tier $t+1$. For each query, we predict the rank of the gold answer entity using quantile regression [Koenker and Hallock, 2001], again on a hold-out set. Given a chosen quantile q and the rank of the gold answer entity $R_i^{(t)}$ at tier t, we train a regressor to predict $\hat{k}_i^{(t)}$ by minimizing

$$\mathcal{L}_q(\hat{k}_i^{(t)}, R_i^{(t)}) = \max\left[q(R_i^{(t)} - \hat{k}_i^{(t)}), (q-1)(R_i^{(t)} - \hat{k}_i^{(t)})\right].$$

As input features to our quantile regressor, we represent the i-th query by its sorted $|\mathcal{V}|$-dimensional score distribution $S_{i1}, \ldots S_{i|\mathcal{V}|}$ from tier t of the cascade, hypothesizing that these score distributions encode uncertainty information correlated to the difficulty of queries. In practice, we implement our regressor as a single-layer MLP trained on half of the dev set, and validated on the remaining dev examples. We will show in our experiments that this approach boosts CascadER's ability to balance effectiveness and efficiency compared to static pruning.

5.5 Evaluation

In this section, we evaluate CascadER on our proposed CoDEx benchmark alongside existing datasets for link prediction. As introduced in Chapter 2.2.1, the evaluation task we consider is ranking-based link prediction. Our evaluation metrics are MRR and hits@k for $k \in \{1, 3, 10\}$. Following the standard in the literature [Bordes et al., 2013, Ruffinelli et al., 2020], we report metrics in the filtered setting, masking out all known answers to test queries other than the gold answer entity in question, in order to avoid false negatives.

5.5.1 Experimental Setup

Data We consider five link prediction benchmarks: Our new datasets **CoDEx-S** and **CoDEx-M**, the **FB15K-237** encyclopedic knowledge benchmark [Toutanova and Chen, 2015], the **WN18RR** linguistic KB extracted from WordNet [Dettmers et al., 2018], and the **RepoDB** KB of

Table 5.9: Statistics of the existing KG link prediction datasets considered in our experiments. We also use CoDEx-S and CoDEx-M, statistics of which are given in Tables 5.4 and 5.5.

	Structure						Avg. text length
	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	# train	# dev	# test	Density	
RepoDB	2,748	1	5,342	667	668	0.00176	55.46
FB15K-237	14,541	237	272,115	17,535	20,466	0.00293	138.95
WN18RR	40943	11	86835	3034	3134	0.00011	13.91

drug-disease treatment relationships [Brown and Patel, 2017, Nadkarni et al., 2021]. For FB15K-237 and WN18RR, we use the linked entity descriptions provided by Wang et al. [2021a]; for RepoDB we use the linked drug and disease descriptions provided by Nadkarni et al. [2021]; for the CoDEx datasets, we use the Wikipedia page extracts as entity descriptions.

Table 5.9 provides structural and textual statistics for RepoDB, FB15K-237, and WN18RR. For all datasets, we use the publicly available standard splits.

Baselines We group our baselines by methodology class:

- **KGE baselines**: We consider the following KGEs, all of which have achieved competitive or state-of-the-art performance on one or more of the datasets we consider: **RESCAL** [Nickel et al., 2011], **TransE** [Bordes et al., 2013], **ComplEx** [Trouillon et al., 2016], and **RotatE** [Sun et al., 2019].

- **LM baselines**: We consider the **StAR dual-encoder** architecture [Wang et al., 2021a] and the **KG-BERT cross-encoder** architecture [Yao et al., 2019], both trained in a multi-task setting with triple classification, margin ranking, and relation classification losses following the literature [Kim et al., 2020]. Note that due to the inference cost of KG-BERT on the larger datasets FB15K-237 and WN18RR (e.g., around one month for FB15K-237 [Kocijan and Lukasiewicz, 2021]), we report results for these datasets from [Kim et al., 2020].

- **Ensemble baselines**: We consider the following full additive ensembling baselines, as defined in § 5.4.1, controlled by a weighting hyperparameter α tuned on the dev set: **KGE + KGE** ensembles the two strongest KGE baselines in terms of MRR on the validation set; **KGE + StAR** [Wang et al., 2021a] ensembles the best KGE with StAR; and **KGE + KG-BERT** [Nadkarni et al., 2021] ensembles the best KGE with KG-BERT.

CascadER variants Our first-tier KGE in CascadER is the best-performing baseline KGE in terms of validation MRR. We set an inference time limit of 2 hours for CoDEx-S and RepoDB,

Table 5.10: CascadER outperforms state-of-the-art single-modality and cross-modality link prediction approaches on FB15K-237 and WN18RR. **Bold + underline**: Best performance. Underline: Second-best performance. OOT refers to out-of-time using our inference time limit of 24 hours for FB15K-237 and 6 hours WN18RR.

	FB15K-237				WN18RR			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RESCAL	0.3559	0.2629	0.3926	0.5406	0.4666	0.4387	0.4797	0.5172
TransE	0.3128	0.2206	0.3473	0.4973	0.2278	0.0531	0.3682	0.5201
ComplEx	0.3477	0.2533	0.3836	0.5359	0.4749	0.4381	0.4898	0.5474
RotatE	0.3333	0.2396	0.3676	0.5218	0.4781	0.4395	0.4941	0.5527
StAR	0.296	0.205	0.322	0.482	0.401	0.243	0.491	0.709
KG-BERT	0.267	0.172	0.298	0.458	0.331	0.203	0.383	0.597
KGE + KGE	0.3630	0.2672	0.4016	0.5535	0.4900	0.4521	0.5016	0.5617
KGE + StAR	0.3643	0.2709	0.3989	0.5522	0.5385	0.4716	0.5645	0.6651
KGE + KG-BERT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
CascadER	**0.3860**	**0.2903**	**0.4231**	**0.5782**	**0.5651**	**0.4756**	**0.6126**	**0.7379**

6 hours for WN18RR, and 24 hours for CoDEx-M and FB15K-237, and search for the optimal cascade on the validation set among the following hyperparameters: The choice of LMs in the cascade (StAR dual-encoder, KG-BERT cross-encoder, or both); candidate pruning strategy (static versus dynamic); quantile $q \in \{0.5, 0.75, 0.9, 0.95\}$; and weighting hyperparameter $\alpha^{(t)} \in [0.05, 0.95]$ at each tier.

Software and hardware We implement all KG embeddings using the open-source LibKGE PyTorch library [Broscheit et al., 2020b]. We use the pretrained KGE checkpoints provided by LibKGE for FB15K-237 and WN18RR. For the other datasets, we follow a similar hyperparameter tuning strategy to that proposed by by Ruffinelli et al. [2020] for tuning our KGE baselines.

We implement all LMs with the Huggingface transformers PyTorch library [Wolf et al., 2020] using the same base language model, which is BERT-BASE [Devlin et al., 2019] for all benchmarks except RepoDB, and PubMedBERT [Gu et al., 2021] for RepoDB. We use the following hyperparameters: Batch size of 16, learning rate of 10^{-5}, and 10 epochs. We use a maximum sequence length of 32, 64, and 256 respectively for WN18RR, RepoDB, and all other datasets.

For all of the ensemble baselines and CascadER, we tune the weighting hyperparameter $\alpha \in [0.05, 0.95]$. For all experiments, we use a single NVIDIA Quadro RTX 8000 GPU with 48 GB of RAM.

Table 5.11: CascadER achieves appreciable accuracy gains over single-modality and cross-modality approaches on our newly proposed datasets CoDEx-S and CoDEx-M. **Bold + underline**: Best performance. Underline: Second-best performance. OOM refers to out-of-memory during training. OOT refers to out-of-time using our inference time limit of 2 hours for CoDEx-S and 24 hours for CoDEx-M.

	CoDEx-S				CoDEx-M			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RESCAL	0.4040	0.2935	0.4494	0.6225	0.3173	0.2444	0.3477	0.4557
TransE	0.3540	0.2185	0.4218	0.6335	0.3026	0.2232	0.3363	0.4535
ComplEx	0.4646	0.3714	0.5038	0.6455	0.3365	0.2624	0.3701	0.4758
RotatE	0.2587	0.1586	0.2916	0.4609	OOM	OOM	OOM	OOM
StAR	0.3540	0.2306	0.4051	0.6007	0.2726	0.1888	0.3042	0.4342
KG-BERT	0.2849	0.1472	0.3310	0.5848	OOT	OOT	OOT	OOT
KGE + KGE	0.4665	0.3712	0.5082	0.6518	0.3466	0.2695	0.3808	0.4925
KGE + StAR	0.4751	0.3717	0.5249	0.6712	0.3554	0.2767	0.3901	0.5064
KGE + KG-BERT	0.4812	**0.3764**	0.5290	**0.6898**	OOT	OOT	OOT	OOT
CascadER	**0.4839**	**0.3764**	**0.5383**	0.6871	**0.3830**	**0.2998**	**0.4221**	**0.5423**

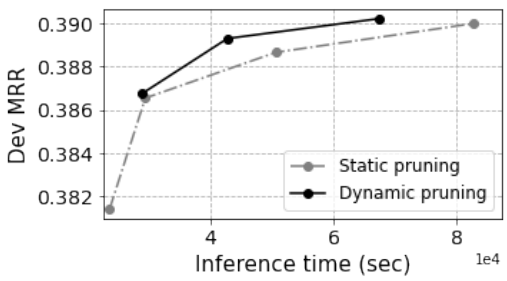
Table 5.12: CascadER achieves state-of-the-art performance on the drug repurposing benchmark RepoDB.

	MRR	H@1	H@3	H@10
RESCAL	0.4351	0.3144	0.4903	0.6767
TransE	0.3472	0.2043	0.3728	0.6400
ComplEx	0.4620	0.3406	0.5225	0.7043
RotatE	0.2971	0.1811	0.4903	0.5314
StAR	0.3472	0.2043	0.4102	0.6400
KG-BERT	0.2991	0.1602	0.3428	0.5996
KGE + KGE	0.4637	0.3398	0.5262	0.7081
KGE + StAR	0.4774	0.3496	0.5434	0.7208
KGE + KG-BERT	0.5101	0.3713	0.5771	0.7799
CascadER	**0.5156**	**0.3817**	**0.5831**	**0.7814**

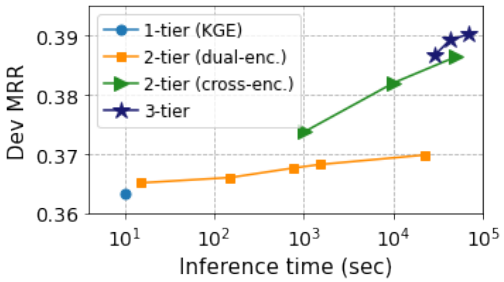
5.5.2 Results and discussion

Table 5.10, 5.11, and 5.12 provide link prediction performance results for FB15K-237 and WN18RR, the CoDEx datasets, and RepoDB, respectively. We observe the following take-aways:

CascadER achieves robust and appreciable gains over baselines across datasets. It outperforms the best KGE by up to 8.70 points MRR (WN18RR) and the best LM by up to 16.84 points MRR (RepoDB), demonstrating that cross-modal ensembling can significantly improve upon single-modality approaches. In fact, we suspect that our reported performance numbers are



(a) Dynamic pruning balances effectiveness and efficiency better than static pruning. We consider a three-tier cascade with pruning only between tiers two and three.



(b) CascadER is effective at any level of cost constraint compared to the base KGE. For the three-tier cascade, we use dynamic pruning only between tiers two and three.

Figure 5.9: Top-left corner is best: Pareto curve analysis on the dev set of FB15K-237. We use quantiles $q \in \{0.5, 0.75, 0.9, 0.95, 1\}$ in our analyses and exclude any quantiles that lead to CascadER exceeding our inference time limit of 24 hours.

lower bounds, as more advanced base KGEs and LMs will likely improve CascadER's performance further.

Full additive ensembling is not necessary to maximize effectiveness. Our KGE + KG-BERT additive ensemble baseline is competitive on CoDEx-S and RepoDB, but it encounters out-of-time errors on FB15K-237, WN18RR, and CoDEx-M. By contrast, CascadER outperforms all baselines across datasets while staying within our cost budgets. This suggests that full additive ensembling is not necessary to achieve the majority of accuracy gains in link prediction, and that cascaded reranking is sufficient.

5.5.2.1 Pareto curve analysis

Next, to drill down into the effectiveness-efficiency tradeoff of CascadER, we provide a Pareto curve analysis. In Figure 5.9, we plot the effectiveness (validation MRR) and efficiency (inference cost in wall-clock time) against CascadER's key hyperparameters, the candidate pruning strategy and the number of tiers. We observe the following:

Dynamic pruning balances effectiveness and efficiency better than static pruning. As shown in Figure 5.9a, dynamic pruning leads to steeper MRR improvements than its static counterpart with comparable inference times.

CascadER is effective at every level of budget constraint. Figure 5.9b compares effectiveness-efficiency curves of two-tiered and three-tiered CascadER structures against the performance of its base KGE. Even the least performant variant of CascadER achieves higher MRR than the KGE

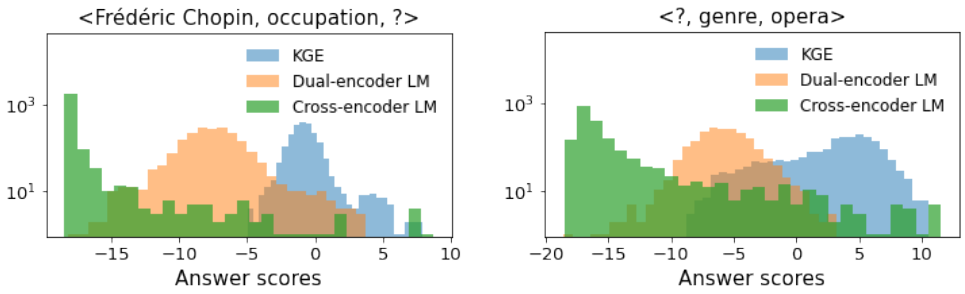


Figure 5.10: The cross-encoder's score distributions are highly skewed left, whereas the score distributions of the KGE and dual-encoder are more normal.

with nearly equivalent inference times.

With a more relaxed budget, cross-encoding is more beneficial than dual-encoding. Figure 5.9b confirms that two-tiered CascadER with a cross-encoder is much more effective than two-tiered CascadER with a dual-encoder. At an inference time of around 1000 seconds, our two-tiered dual-encoder and cross-encoder architectures achieve 0.3683 and 0.3738 MRR respectively. These results suggest that *very little* reranking with a cross-encoder is often more beneficial than a *large amount* of reranking with a dual-encoder. Our findings are consistent with the information retrieval literature, which have consistently shown that cross-encoders are strong rerankers [Matsubara et al., 2020, Xiong et al., 2020a, Luan et al., 2021].

5.5.2.2 Qualitative analysis

Finally, to illuminate the benefits of cross-modal reranking, we provide a brief qualitative analysis of how cross-modal ensembling exploits the complementary behaviors of structure and text models. Figure 5.10 plots the empirical answer score distributions to two link prediction queries from CoDEx-M. We observe that the score distributions of the KGE and the dual-encoder are more normally shaped, whereas the cross-encoder's score distribution is highly skewed left. This suggests that cross-encoders more aggressively filter out irrelevant candidate answers to queries, perhaps due to their ability to capture term overlap (or lack thereof) between text pairs with relatively high precision compared to vector matching models that do not use cross-attention [Luan et al., 2021]. Additive reranking between a KGE and a cross-encoder appears to have a highly beneficial effect due to these distributional differences: Additive re-weighting with a cross-encoder helps widen the margins or score gaps the most between the scores of gold answers and those of negative candidates, as shown in Figure 5.11.

Beyond complementary scoring behavior, we observe another aspect of complementarity in our results, which is that we achieve the largest gains over the base KGE on the sparsest graphs. As

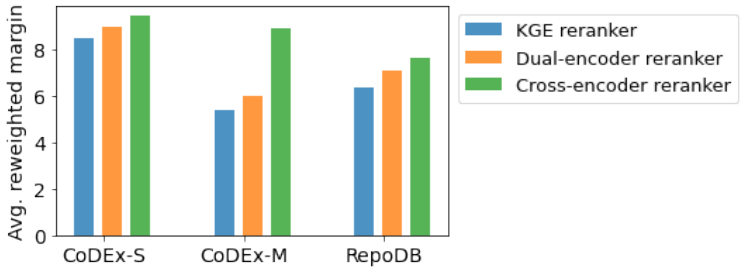


Figure 5.11: Additive LM reranking over a base KGE helps widen the score gaps between gold answers and negative candidates the most. We show the average score gap between the gold answer to each query and all negative candidates for the three datasets on which full cross-encoding was computationally feasible. For each dataset, the base ranker is the best KGE in terms of validation MRR, and the second-tier KGE reranker is the second-best KGE in terms of validation MRR.

Table 5.13: CascadER provides larger gains on sparser graphs: MRR comparison between the best KGE baseline and CascadER on each dataset.

Density	CoDEx-S 1×10^{-2}	FB15K-237 2×10^{-3}	CoDEx-M 1×10^{-3}	RepoDB 1×10^{-3}	WN18RR 1×10^{-4}
Base KGE	0.4646	0.3559	0.3365	0.4620	0.4781
CascadER	0.4839	0.3860	0.3830	0.5156	0.5651
CascadER improvement	**+1.93**	**+3.01**	**+4.65**	**+5.36**	**+8.70**

illustrated in Table 5.13, CascadER improves MRR by 2-3 points over the best KGE on CoDEx-S and FB15K-237, our two densest KGs. By contrast, on CoDEx-M, WN18RR, and RepoDB, CascadER improves MRR by 5-9 points over the best KGE baseline. The gain is the largest on WN18RR, which, at a density on the order of 10^{-4}, is the sparsest graph considered in our experiments. These results confirm that text provides an especially useful source of auxiliary information when the relational structure of the data is highly sparse.

5.6 Conclusion

In this chapter, we considered the task of automatically completing factual KBs, formulated as a link prediction ranking problem in a multi-relational graph. While there is a large body of related literature on structure-only link prediction, we noted the relative lack of joint structure and text approaches for this task, even though many KBs are linked to rich side textual attributes. To provide a foundation for future work in this direction, we made two key contributions. The first, CoDEx, is a new cross-modal KBC benchmark derived from Wikidata and Wikipedia. We demonstrated the advantages of CoDEx over a comparable existing benchmark from both quantitative and qualitative perspectives, emphasizing its difficulty for the link prediction task using structure-only approaches

and the interpretability of its content.

Next, we introduced CascadER, a cross-modal multi-stage cascade approach that combines structure-only KB embeddings with language models for link prediction in a sequential reranking architecture. We showed that CascadER achieves state-of-the-art performance on multiple link prediction benchmarks, including CoDEx, by effectively combining structure and text models while maintaining efficiency. We provided analysis to understand the advantages that our cross-modal approach provides over single-modality approaches. CascadER's promise in this task suggests its versatility across many different datasets and problem settings, making it an ideal model for combining structure and text to discover novel relationships between entities.

Part II

Document Interaction and Content Mining

CHAPTER 6

Discovering Activities in Personal Information Collections

The material in this chapter is derived from the paper "Toward Activity Discovery in the Personal Web" [Safavi et al., 2020], which appeared in the proceedings of the 13th ACM Conference on Web Search and Data Mining (WSDM) in 2020.

6.1 Introduction

Having considered text-augmented graph learning for world knowledge representation, we now consider more diverse information retrieval settings that require modeling structured interactions and textual content attributes. This central research question addressed by this chapter is motivated by the vision that scientist Vannevar Bush set out in his influential 1945 essay *As We May Think*. Bush described a device called a "memex," in which "an individual stores all his books, records, and communications, and...may be consulted with exceeding speed and flexibility," and motivated his memex by the *associative* nature of the human mind: "With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts...[it] is awe-inspiring beyond all else in nature" [Bush, 1945]. Our goal is to expand on the idea of *associating individuals' personal information items*—what Bush called books, records, and communications, and what today might be files, emails, and messages—according to their higher-level purposes and usages. Specifically, our central research question is: Can we model activities (projects, hobbies, tasks) in individuals' heterogeneous personal information collections (files, emails, contacts, etc) using both interaction structure and document contents to power personal search and recommendation?

The goal of activity discovery is to help people better organize, retrieve, and utilize their personal information. For example, modern email clients support tagging and foldering, but individuals struggle to maintain these efforts because manual curation is costly [Whittaker and Sidner, 1996, González and Mark, 2004, Whittaker et al., 2011, Grevet et al., 2014]. We envision next-

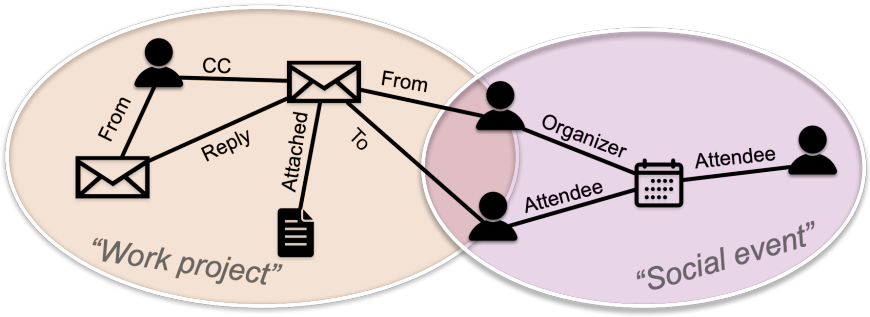


Figure 6.1: A Personal Web consisting of two activities.

generation email clients that automatically learn users' ongoing activities to organize or even prioritize emails and meetings based on those activities. Semantic and conversational search systems can also benefit from inferring users' activities. For example, such systems could allow users to directly search by concept or activity (e.g., "Show me all receipts related to my home remodel"), without requiring users to explicitly specify their activities. Even helping people understand how they spend their time by activity can be useful for productivity-related reflection and planning.

However, this direction comes with unique challenges. For one, people's activities are complex and fluid. They can exist on varying time scales and evolve over time. Some activities overlap with, or subsume, one another. Ideally, automated approaches to personal activity discovery should capture such complexity. Another challenge is that of evaluation, which is difficult due to (well-founded) privacy concerns, a lack of standardized methodology, and the high cost of obtaining explicit feedback [Zamani et al., 2017].

6.1.1 Contributions

Guided by the concept of "associations between items," we learn representations of personal information collections such that objects related by activity have similar representations and can be directly compared regardless of type. We leverage the inherent *graph structure* of personal information collections, augmenting items with interaction- and structure-based links to form what we call a Personal Web. We propose a fast linear representation learning method over the Personal Web to obtain and incrementally update object representations. In extensive experiments across multiple evaluation tasks, we observe the gains of Personal Webs over various competitive baselines in terms of retrieval metrics over personal information collections.

Our contributions are as follows:

- **Problem definition** (Ch. 6.3): We propose the problem of activity discovery in heterogeneous personal information collections, toward helping people better manage and search over their personal information.

- **Interaction and content model** (Ch. 6.4): We propose to model individuals' personal information collections as Personal Webs. We propose to learn unsupervised representations of entities in Personal Webs, devising an objective that incorporates both interaction and content in Personal Webs. We derive efficient, exact techniques to update representations as new data arrive, up to $470\times$ faster than learning from scratch.
- **Extensive evaluation** (Ch. 6.5, 6.6, and 6.7): We conduct a range of experiments. We first gather human judgments for activity-specific relationships surfaced by our model in a unique, small-scale user study. We next devise an extrinsic email recipient recommendation task to test the versatility of our representations. In both tasks, our representations outperform competitive baselines across a variety of information retrieval metrics. We also evaluate our model's offline and online scalability on a large real email dataset and demonstrate that learning our model is hundreds of times faster than baselines in the offline and online settings.

6.2 Related Work

Personal information management Personal information management (PIM) addresses how people organize and retrieve items from their personal information collections [Teevan et al., 2006]; for brevity, we refer the reader to [Jones et al., 2017] for a comprehensive overview of the literature. A representative, highly influential example is *Stuff I've Seen* (SIS) [Dumais et al., 2003], a unified search index aggregating heterogeneous personal information objects from users' desktops. SIS was followed by *Implicit Query* [Dumais et al., 2004], an email plug-in that automatically displays desktop items related to the user's current email of focus. These early works motivate our goal of unifying heterogeneous personal information objects, as they provide empirical evidence that people prefer such unified systems [Dumais et al., 2003].

Recent studies develop privacy-preserving machine learning approaches for PIM, most often for email [Bendersky et al., 2017, Zamani et al., 2017, Zhao et al., 2018]. Such studies share our high-level goal of helping individual manage their personal information in a private and efficient manner. However, as far as we are aware, we are the first to specifically consider representation learning over heterogeneous personal information collections, and moreover the first to address the problem by proposing a unified interaction and content model.

Activity inference The literature on inferring individuals' activities over personal information items focuses mostly on email [Dredze et al., 2006, Dredze and Wallach, 2008, Qadir et al., 2016]. Dredze et al. [2006] devise a supervised method to classify emails into activities by calculating

overlap statistics among email messages and known activities. More recently, Qadir et al. [2016] learn activities from workplace email with an unsupervised generative model that considers contexts like subject line, recipients, and linguistic features of emails. A major difference between these works and our own is that we wish to discover activities that span multiple *types* of objects, including but not limited to email. From a problem setting perspective, the *TaskTracer* system [Shen et al., 2006] for activity management over heterogeneous desktop items is more related. However, *TaskTracer* is supervised and requires users to enforce a hierarchy over their data, whereas we take an unsupervised approach.

Graph similarity search Because we model personal information collections as graphs, our task can be cast as a similarity search problem among representations of nodes in a graph [Perozzi et al., 2014, Tang et al., 2015, Grover and Leskovec, 2016, Hamilton et al., 2017, Hamilton, 2020]. The most topically-related approach of which we are aware learns node representations in a graph representing professional email communications between people [Jin et al., 2019b]. However, the node embeddings proposed by Jin et al. [2019b] are tailored to the task of inferring people's roles in professional hierarchies, whereas we consider diverse personal information management tasks. We also formulate our problem and propose a model that handles arbitrary heterogeneous objects rather than solely emails or people. While vector representations of heterogeneous multi-relational graphs have been previously proposed [Dong et al., 2017, Wang et al., 2019d], these approaches require the heuristic construction of various semantic paths in the graph ("metapaths" [Sun et al., 2011]), which we do not require in our approach.

6.3 Problem Definition

According to Dredze et al. [2006], activities are collaborative practices that have state and a goal (e.g., *organizing a conference*). Using this definition, we state the following desiderata for activity discovery over heterogeneous personal information collections:

- **Privacy-preserving**: Personal information collections contain sensitive private information [Zhao et al., 2018]. To avoid data leakage, our model should operate on a per-person basis, directly on personal devices, without leveraging collective patterns across users.
- **Requires no supervision**: Manually organizing personal data into, e.g., social circles or email folders, requires a nontrivial amount of effort [Whittaker et al., 2011, Mcauley and Leskovec, 2014]. Accordingly, we do not assume the presence of labeled data, although an ideal model should be able to incorporate activity labels if available.

- **Online updates**: Individuals' activities naturally change over time. To handle this evolution, we should be able to incrementally and efficiently update our model as new data arrive.

We state our research problem as follows: Given an entity e from an individual's heterogeneous personal information collection C, find entities e' that are "related" or "associated" to e in the context of u's activities in a manner that (1) preserves privacy by being learned on an individual, on-device basis, (2) is unsupervised, and (3) can operate in an online setting.

6.4 Methodology

We now introduce our joint interaction and content learning approach over heterogeneous personal information collections.

6.4.1 Personal Webs Overview

As our problem statement is open-ended, several classes of methodology could be employed. For example, a retrieval approach would rank the most related entities e' to a query entity e in terms of activities, whereas a clustering approach would directly group entities into activities. To handle this open-endedness we propose to learn versatile representations of heterogeneous personal information items that can be used in numerous settings like retrieval, clustering, and classification.

Our approach naturally combines interaction and content, relying on two ideas. The first is that personal information objects have inherent graph structure (Figure 6.1). The second is that closely-connected objects in the graph sharing textual content indicative of activities should have similar representations.

Personal Webs Given an individual's personal information collection $\mathcal{P}$, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from $\mathcal{P}$. In this chapter we focus on unweighted and undirected graphs, but our model can be straightforwardly extended to handle edge weights and directionality. We call $\mathcal{G}$ the **Personal Web** associated with that individual.

Each entity (node) in $\mathcal{V}$ has an associated type, such as *Email*, *Calendar Appointment*, or *Contact*. Each node may be also associated with textual content, for example email subject lines, bodies, etc. That said, because we consider arbitrarily heterogeneous objects, not all nodes are documents (e.g., *Contacts*, *Photos*, etc). We do not include the individual who owns the data in the Personal Web, since they are implicitly "connected" to all other nodes in the graph.

Each edge in $\mathcal{E}$ encodes a semantically meaningful relationship between entities. For example, an edge connecting a *Calendar Appointment* to a *Contact* might signify that the appointment was

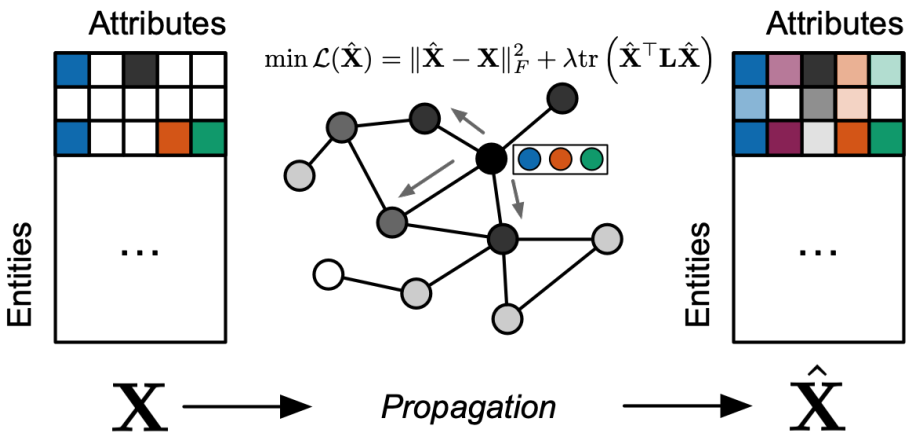


Figure 6.2: Joint learning over interaction and content in the Personal Web.

organized or attended by that person. In practice, a Personal Web can be instantiated in many different ways, the effects of which we illustrate in Ch. 6.4.2.

6.4.2 Entity Representation Learning

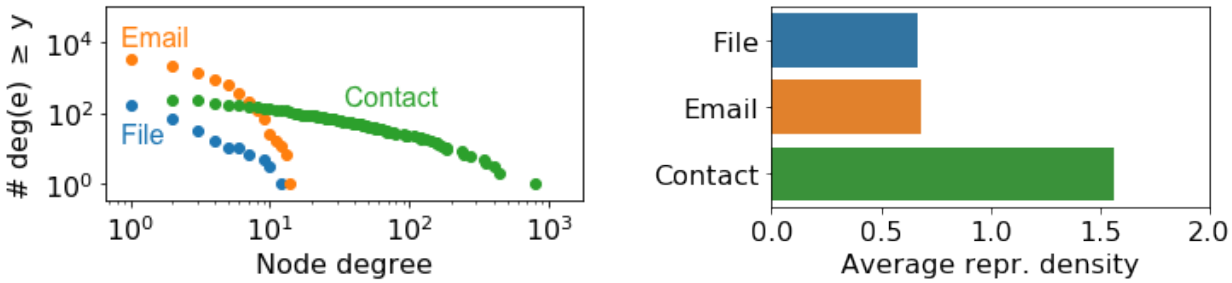
We learn entity representations with a propagation process over $\mathcal{G}$ that yields similar representations for entities that are closely-connected in $\mathcal{G}$ and/or share similar features. The propagation starts from a set of seed entities in $\mathcal{G}$ that are associated with "activity-specific" textual content features. In the following sections, we will provide concrete examples of such features.

Objective Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a matrix in which nonnegative entry X_{ij} corresponds to entity i's membership strength in activity-indicating textual attribute j. We propagate the seeds' attribute membership strengths X_{ij} across the graph to learn $\hat{\mathbf{X}} \in \mathbb{R}^{n \times p}$, the entity-attribute membership matrix for all entities in $\mathcal{G}$, by minimizing the loss function

$$\mathcal{L}(\hat{\mathbf{X}}) = \left\|\hat{\mathbf{X}} - \mathbf{X}\right\|_F^2 + \lambda \, \text{tr}\left(\hat{\mathbf{X}}^\top \mathbf{L}\hat{\mathbf{X}}\right), \tag{6.1}$$

where the graph's Laplacian is given as $\mathbf{L} = \mathbf{D} - \mathbf{A}$ for diagonal degree matrix $\mathbf{D}$ and symmetric adjacency matrix $\mathbf{A}$.

The first term in (6.1) constrains the learned attribute values for seed entities to be close to their initial values. The second term enforces graph smoothness and yields similar learned attribute-value distributions for linked entities, controlled by regularization hyperparameter λ. Note that (6.1) recalls label propagation and convolutions over graphs [Zhu, 2005, Kipf and Welling, 2017], but such techniques are typically used to predict node classes, which is not our



(a) Log-log cumulative degree distribution in the Personal Web, stratified by type.

(b) Average representation density, using noun phrases as textual attributes, stratified by type.

Figure 6.3: Capturing semantic differences in entity types through representation density for a random *Medium* inbox from the Avocado dataset (Table 6.3). On average, *Contact* nodes have the highest degrees and thus the densest representations, reflecting that people usually participate in more activities than other types of entities.

goal. Rather, we use propagation to obtain entity representations: The i-th entity's final representation is given as $\hat{\mathbf{x}}^i$, the i-th row of $\hat{\mathbf{X}}$.

Note that we learn a separate model per person without sharing any information across people. All textual attributes in a Personal Web therefore come directly from the owning individual's personal information objects.

Entity attributes The propagation seed matrix $\mathbf{X}$ maps select entities to textual content features that directly or indirectly indicate activities. In the unsupervised case, which is the focus of this work, we consider noun phrase frequencies and latent topic memberships (LSA [Deerwester et al., 1990] and LDA [Blei et al., 2003]) as attributes and their associated strengths. The motivation is that noun phrases often directly correspond to project, task, or goal names [Benetka et al., 2019], whereas latent topics capture semantic relatedness among groups of documents. Importantly, when noun phrases are used, our approach can produce fully human-interpretable representations, since the columns in $\mathbf{X}$ correspond to natural language. It also naturally handles semi-supervision: We can consider activity labels as attributes if available, although we leave this direction for future work. although we focus on the (arguably more realistic) unsupervised setting in this chapter. That said, it has been shown that users struggle to impose hierarchy over their personal data and settle on a single meaning of a given label [Whittaker and Sidner, 1996, Kulesza et al., 2014]. Therefore, we let users' activities "bubble up" without supervision rather than requiring labels.

Entity type semantics Our representation approach implicitly captures entity type semantics via *representation density*. Figure 6.3 illustrates this effect with a graph from the Avocado dataset (Ch. 6.6) consisting of the following relations: *Contact-Email*, signifying who sent or received a

specific email; *Email-Email*, signifying direct replies on email threads; and *File-Email*, signifying files attached to emails. Figure 6.3a demonstrates that a node's degree corresponds strongly with its type: The degrees of *Contact* entities are up to orders of magnitude larger than those of *Email*s and *File*s.

Figure 6.3b demonstrates that *Contact* entity representations are on average twice as dense as *Email* and *File* entity representations, due to their having higher degree. In this case, the higher density of *Contact* entities reflects the idea that people usually participate in more activities than do single emails or files. This effect is an advantage of our approach: We allow entities' representation densities to vary according to their type semantics, while still representing all heterogeneous objects in a shared vector space.

6.4.2.1 Offline Learning

We are given a graph $\mathcal{G}$ and its corresponding Laplacian matrix $\mathbf{L}$. To find the entity-attribute membership matrix $\hat{\mathbf{X}}$, we first take the derivative of (6.1) with respect to $\hat{\mathbf{X}}$:

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{X}}} = 2\left(\hat{\mathbf{X}} + \lambda \mathbf{L}\hat{\mathbf{X}} - \mathbf{X}\right).$$

Setting the derivative to 0 and solving for $\hat{\mathbf{X}}$, we obtain

$$\hat{\mathbf{X}} = \left(\mathbf{I} + \lambda \mathbf{L}\right)^{-1} \mathbf{X}, \tag{6.2}$$

where $\mathbf{I}$ is the identity matrix. In practice, to avoid a computationally prohibitive matrix inversion, we solve for each column of $\hat{\mathbf{X}}$ with Jacobi iteration, which is guaranteed to converge because the matrix inverted in (6.2) is diagonally dominant:

$$\hat{\mathbf{x}}_i^{(j+1)} = \left(\mathbf{I} + \lambda \mathbf{D}\right)^{-1} \left(\mathbf{x}_i + \lambda \mathbf{A}\hat{\mathbf{x}}_i^{(j)}\right), \tag{6.3}$$

where $\mathbf{x}_i$ is the i-th column of $\mathbf{X}$, and $\hat{\mathbf{x}}_i^{(j)}$ is the i-th column of $\hat{\mathbf{X}}$ in the j-th iteration.

6.4.2.2 Online Learning

In a more realistic setting, we wish to incrementally update entity representations $\hat{\mathbf{X}}$ as new data arrive over time. We make the mild assumption that as new data arrive, entities' textual attributes and their strengths can be obtained on-the-fly rather than requiring a full pass over the data. In practice, this assumption holds when we take noun-phrases as attributes, but not when we require a decomposition of the full document-term matrix (e.g., LSA, LDA).

The online setting consists of two cases:

- **Case 1**: Only the graph structure changes, meaning a new personal information object arrives (e.g., an email) and/or a new link between objects arrives (e.g., an email is forwarded to a colleague), without observing new textual attributes.

- **Case 2**: Both the graph structure and the graph's textual attributes change, meaning in addition to structural changes in $\mathcal{G}$ we observe new textual information, like an unseen noun phrase.

We begin with **Case 1**. Given newly observed edge (i, j), let $\Delta\mathbf{D}$ and $\Delta\mathbf{A}$ represent rank-one updates to $\mathbf{D}$ and $\mathbf{A}$, respectively, such that the updated degree matrix is $\mathbf{D}_{\text{new}} = \mathbf{D} + \Delta\mathbf{D}$, the updated adjacency matrix is $\mathbf{A}_{\text{new}} = \mathbf{A} + \Delta\mathbf{A}$, and the updated Laplacian is

$$\mathbf{L}_{\text{new}} = \mathbf{D}_{\text{new}} - \mathbf{A}_{\text{new}} = (\mathbf{D} - \mathbf{A}) + (\Delta\mathbf{D} - \Delta\mathbf{A}) = \mathbf{L} + \Delta\mathbf{L}. \tag{6.4}$$

Because only the graph structure changes, the online objective matches (6.1), except that $\mathbf{L}_{\text{new}}$ replaces $\mathbf{L}$. Therefore, following (6.2), the closed-form solution of the online objective is

$$\hat{\mathbf{X}}_{\text{new}} = (\mathbf{I} + \lambda\mathbf{L}_{\text{new}})^{-1}\,\mathbf{X} = (\mathbf{I} + \lambda\mathbf{L} + \lambda\Delta\mathbf{L})^{-1}\,\mathbf{X}. \tag{6.5}$$

Equation (6.5) can be naively solved with Jacobi iteration as in Ch. 6.4.2.1, but we devise a faster approach that reuses previous computation *and* arrives at the same solution as our offline model. The key to efficiency is that both $\Delta\mathbf{D}$ and $\Delta\mathbf{A}$ are rank one and can be expressed as outer products. Letting $\mathbf{e}_i \in \mathbb{R}^n$ be an indicator vector with its i-th entry equal to one, and zero elsewhere, we can express $\Delta\mathbf{D} = \mathbf{e}_i\mathbf{e}_i^\top + \mathbf{e}_j\mathbf{e}_j^\top$ and $\Delta\mathbf{A} = \mathbf{e}_i\mathbf{e}_j^\top + \mathbf{e}_j\mathbf{e}_i^\top$. It then follows that

$$\begin{aligned}\Delta\mathbf{L} &= \Delta\mathbf{D} - \Delta\mathbf{A} \qquad \rightarrow \text{ From (6.4)} \\ &= (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top \end{aligned} \tag{6.6}$$

is also rank one. Using the Sherman-Morrison formula for rank-one updates to a matrix inverse [Sherman and Morrison, 1950], it follows that

$$\begin{aligned}\hat{\mathbf{X}}_{\text{new}} &= (\mathbf{I} + \lambda\mathbf{L} + \lambda\Delta\mathbf{L})^{-1}\mathbf{X} \qquad \rightarrow \text{From (6.5)} \\ &= \left(\mathbf{I} + \lambda\mathbf{L} + \lambda(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top\right)^{-1}\mathbf{X} \qquad \rightarrow \text{ From (6.6)} \\ &= \left((\mathbf{I} + \lambda\mathbf{L})^{-1} - \frac{(\mathbf{I} + \lambda\mathbf{L})^{-1}\lambda(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top(\mathbf{I} + \lambda\mathbf{L})^{-1}}{1 + (\mathbf{e}_i - \mathbf{e}_j)^\top(\mathbf{I} + \lambda\mathbf{L})^{-1}\lambda(\mathbf{e}_i - \mathbf{e}_j)}\right)\mathbf{X} \\ &= \hat{\mathbf{X}} - \frac{(\mathbf{I} + \lambda\mathbf{L})^{-1}\lambda(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top\hat{\mathbf{X}}}{1 + (\mathbf{e}_i - \mathbf{e}_j)^\top(\mathbf{I} + \lambda\mathbf{L})^{-1}\lambda(\mathbf{e}_i - \mathbf{e}_j)},\end{aligned}$$

where on the last line we substitute $(\mathbf{I}+\lambda\mathbf{L})^{-1}\mathbf{X} = \hat{\mathbf{X}}$ as per the closed form in (6.2). Now, letting

$$\mathbf{u} = (\mathbf{I} + \lambda\mathbf{L})^{-1}\lambda(\mathbf{e}_i - \mathbf{e}_j) \tag{6.7}$$

and

$$\mathbf{v}^\top = \frac{(\mathbf{e}_i - \mathbf{e}_j)^\top \hat{\mathbf{X}}}{1 + (\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{u}}, \tag{6.8}$$

we can write the rank-one update to $\hat{\mathbf{X}}$ as

$$\Delta\hat{\mathbf{X}} = \hat{\mathbf{X}}_{\text{new}} - \hat{\mathbf{X}} = -\mathbf{u}\mathbf{v}^\top, \tag{6.9}$$

meaning that per newly observed edge, we only need to update $\hat{\mathbf{X}}$ with the outer product in (6.9).

In **Case 2**, where the new edge (i, j) contains at least one entity with previously unobserved textual attributes, we propagate these new attributes across the graph using (6.3) and add the result, along with the result of (6.9), to $\hat{\mathbf{X}}$. In the following sections, we demonstrate both theoretically and empirically that this outer product formulation results in orders of magnitude performance improvement.

6.4.3 Complexity Analysis

To conclude the discussion of our representation learning approach, we discuss computational complexity. Recall that we define a Personal Web as a graph $\mathcal{G}$ consisting of nodes $\mathcal{V}$ and edges $\mathcal{E}$. Here, we use p to refer to the number of textual attributes in the graph.

Offline setting In the offline setting, solving for each column of $\hat{\mathbf{X}}$ requires $O(|\mathcal{E}|)$ time for a fixed number of Jacobi iterations, since the matrix to be inverted in (6.2) has $O(|\mathcal{E}|)$ nonzero entries. Thus, the total complexity is $O(p|\mathcal{E}|)$.

Online setting In the online setting, solving for $\mathbf{u}$ takes $O(|\mathcal{E}|)$ time, again using the Jacobi method. Computing $\mathbf{v}$ takes $O(p|\mathcal{V}|)$ time for $\hat{\mathbf{X}} \in \mathbb{R}^{n\times p}$, and taking the outer product $\mathbf{u}\mathbf{v}^\top$ is also $O(p|\mathcal{V}|)$ for $\mathbf{u} \in \mathbb{R}^{|\mathcal{V}|}$ and $\mathbf{v} \in \mathbb{R}^p$. Therefore, the total complexity of evaluating (6.9) without observing a new entity is $O(|\mathcal{E}| + p|\mathcal{V}|)$.

If a new entity e is observed, according to our offline analysis the complexity is $O(p_e|\mathcal{E}|)$ where p_e is the number of textual attributes for entity e, usually small in practice. Note that we must add new rows and columns to $\mathbf{L}$ and $\mathbf{X}$ as necessary, but this operation scales with the number of

nonzero matrix elements when implemented with sparse matrices. Therefore, the total complexity is $O(|\mathcal{E}| + p|\mathcal{V}| + p_e|\mathcal{E}|) = O(p_e|\mathcal{E}| + p|\mathcal{V}|)$.

6.5 Human Evaluation

Our first mode of evaluation is intrinsic: We obtain activity relatedness judgments from people on their own data, which is crucial because such judgments are nuanced and depend on knowing context and lived experience [Dumais et al., 2003]. Though limited to a small set of individuals, our intrinsic evaluation allows us to directly characterize our approach. We will complement it with a large-scale extrinsic evaluation in the following section by measuring downstream task performance on a public dataset in lieu of direct judgments on private data.

6.5.1 Data

Collection and preprocessing We developed an on-device logging application that all task participants installed on their primary work computers at least two days prior to the rating task. The application indexes all emails and calendar appointments previously downloaded to the participant's machine, and further records the participant's interactions with these and other personal information items on her desktop. Metadata of these items include, e.g., the people associated with an email, the textual content of a file, when an individual clicked on a meeting, how long she focused on a web page, etc. Importantly, all logs are stored locally, the logging tool does not upload any information to the cloud, and all evaluation scripts using these logs were run locally on participants' computers from a USB drive. We collected only aggregate task performance metrics from each participant.

For preprocessing, we discard placeholder emails and appointments (e.g., "automatic reply"), emails and appointments from senders that the participant never personally contacted, emails without the participant on the *To*, *From*, or *CC* lines, emails that the participant only sent to herself, and, following [Qadir et al., 2016], emails and appointments with over 10 recipients (i.e., email blasts). In total, we filtered out 30-70 percent of emails in each raw Personal Web graph with these preprocessing steps. To capture a rough notion of "importance", we retain only web documents/files that the participant dwelled on for at least 10 consecutive seconds. As textual attributes, we extract noun phrases from email/appointment subject lines and document/file titles. We remove general and domain-specific stopwords (e.g., filename extensions like "pdf", email abbreviations like "fwd") and phrases that often appear in search results ("Google Search").

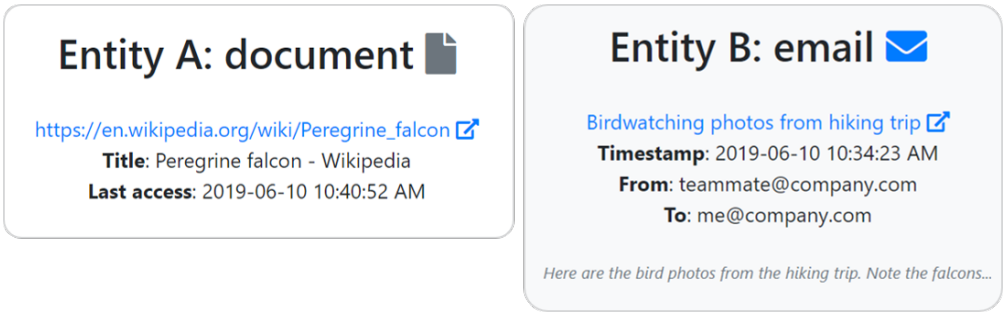


Figure 6.4: Mockup of fictitious entity cards for the human judgment task. Each entity pair is associated with two questions.

Personal Web construction Each Personal Web consists of *Email*, *Calendar Appointment*, *Web Document*, *File*, and *Contact* entities. We define the following semantic relationship types: **(1)** *Contact-Email*, connecting people to emails that they sent, received, or were CC'ed on; **(2)** *Contact-Calendar Appointment*, connecting people to calendar appointments that they organized or attended; **(3-4)** *Email-Web Document* and *Calendar Appointment-Web Document*, connecting emails and appointments to web documents if the participant accessed the document immediately after reading the email or appointment (e.g. when clicking a link in the email body); **(5-6)** *Email-File* and *Calendar Appointment-File*, connecting emails and appointments to desktop files if the participant accessed the document immediately after reading the email or appointment; **(7)** *Email-Email*, connecting pairs of emails that appeared consecutively in a thread (i.e., replies). We construct each Personal Web from the participant's two most recent months of data for ease of contextualization.

6.5.2 Experimental Setup

Participants We recruited $n = 10$ participants (5 female, 5 male, ages 18-54) from a large enterprise technology company. Participants **P1**-**P4** were interns, participants **P5**-**P8** were software engineers, and participants **P9** and **P10** were senior researchers or research managers. Each participant was paid $25 upon completion of the task, which took about 30 minutes on average. The number of entities in each participant's Personal Web is given in Table 6.2.

Task We devise a transductive link prediction task (Chapter 2.2.1) for which our participants provide us relevance labels. We present participants with pairs of their own personal information objects (nodes) and ask them to relate those pairs of objects in the context of their activities (semantic links). Per system to be evaluated, we identify pairs of related objects by following an information retrieval approach: Given a randomly sampled entity e (the "query"), the system con-

structs an ordered list of $k \leq 5$ candidate entities $e' \neq e$ it predicts are "related" or linked in terms of the participant's activities.

Participants performed the task through a locally hosted web application, the interface of which is demonstrated in Figure 6.4. To fit the allotted time of the task, as each pair took 30-45 seconds to rate, we limited the number of ranking systems and query entities such that each participant rated up to 60 pairs: 3 query entities $e \times$ up to 5 candidate related entities $e' \times 4$ ranking systems, which we discuss later in this section.

Questionnaire Each pair of entities displayed to the participant is associated with two questions:

- **Question 1** asks *how* the pair of items are related ("Why do you think the system related this pair of entities?"). The answer choices are: **(1)** *Low-level*, "These entities correspond to the same short-term task, appointment, or goal (e.g., a meeting, a TODO)"; **(2)** *Mid-level*, "These entities correspond to the same long-term project or activity (e.g., a research project, a home remodel)"; **(3)** *High-level*, "These entities correspond to the same general life category, not necessarily with defined start or end dates (e.g., *Personal*, *Professional*, *School*)"; **(4)** *Other*, "These entities are related for reasons not listed above"; **(5)** *Not related*, "The system is wrong. I cannot find any relationship between these entities"; and **(6)** *Unsure*, "The system may have its reasons, but I don't recognize one or more of these entities".
- **Question 2** asks the participant to assess the degree of activity-specific "relatedness" of the displayed pair ("In your opinion, how related is this pair of entities?"). The choices form a graded scale, scored as follows: *Strongly related* (4 points), *related* (3 points), *somewhat related* (2 points), *a little related* (1 points), and *not related at all* or *unsure* (0 points).

Performance metrics Per participant we collect only aggregate system performance metrics from the judgment task. Given a threshold of relatedness from the participant's answers to Question 2 (e.g., *a little related* up to *strongly related*), we compute recall and precision@k. When computing recall, we form the gold set of related entities by pooling judgments across methods and baselines.

Baselines Due to the allotted time and cost of the task, and the fact that all scripts ran locally on personal machines, we restricted the task to two variants of our approach and two promising baselines. We select our baselines out of a number of approaches from retrieval, clustering, and embedding based on their performance in an independent pilot study involving six participants:

- **People Overlap** [Dredze et al., 2006]: For each query entity e, we compute the Jaccard similarity between the people involved in e and all other entities e' excluding the participant,

then rank the entities e' by similarity score and return the top k.

- **node2vec** [Grover and Leskovec, 2016]: We choose node2vec, which is based on the word2vec architecture [Mikolov et al., 2013], among other graph embeddings for its widespread usage and lightweight time and space complexity, which made it feasible to run in a short time on participants' personal devices. For this baseline we first augment each participant's Personal Web by including nodes representing noun phrases and edges connecting entities to the noun phrases that they contain, similar to [Tang et al., 2015, Bendersky et al., 2017]. We then apply node2vec on the augmented graph using its default parameter settings [Grover and Leskovec, 2016]. For each query entity e, we find its k nearest neighbor entities e' in embedding space.

Personal Webs variants The variants of our Personal Webs approach rely on different activity-related textual attributes:

- **NP**: Seed entities' attributes are the noun phrases they contain, and their respective strengths are their frequencies. Note that we do not differentiate between the offline and online versions of our representations for this variant, since they yield the same results.
- **LSA [Deerwester et al., 1990]**: As attributes we take the rank-32 SVD of the document-phrase frequency matrix, treating each dimension of the decomposed matrix as a topic. We set the decomposition rank to 32 based on the results of our independent pilot study, in which we observed that this value led to the best tradeoff of effectiveness and efficiency among a choice of $\{10, 32, 64\}$.

We set $\lambda = 10^2$ for both variants to emphasize the graph structure. After learning representations $\hat{\mathbf{X}}$, we return query entity e's k nearest neighbors e' in vector space, ranked in increasing order of Euclidean distance to the query entity e's vector representation.

6.5.3 Results and Discussion

Table 6.1 gives performance metrics averaged across all study participants. The first group of rows in Table 6.1 uses a permissive binarization of the "relatedness" scale (Question 2), considering pairs to be related unless the participant responded *unsure* or *not related at all*. The second group of rows uses a stricter binarization of relatedness, considering pairs to be related only if the participant chose *strongly related*. It is evident from Table 6.1 that all systems are able to identify related entities when the definition of "relatedness" is relatively loose. Our LSA representations mostly perform best here except for lower ranks of precision, where People Overlap performs best. However, with a stricter definition of relatedness, our NP representations mostly perform best.

Table 6.1: Average performance metrics per system averaged across all participants. Highest score among systems per metric shaded. Top group of rows: Averages across all pairs rated by participants as *a little related* or above. Bottom group of rows: Averages for pairs rated as *strongly related* only.

	***A little related*, *somewhat related*, *related*, and *strongly related* pairs**					
	Recall	Prec@1	Prec@2	Prec@3	Prec@4	Prec@5
People Overlap	0.450 ± 0.11	0.933 ± 0.13	0.933 ± 0.11	0.922 ± 0.11	0.933 ± 0.10	0.933 ± 0.11
node2vec	0.440 ± 0.10	0.900 ± 0.21	0.867 ± 0.16	0.844 ± 0.17	0.858 ± 0.14	0.867 ± 0.14
Personal Webs-NP	0.444 ± 0.07	0.967 ± 0.10	0.933 ± 0.11	0.911 ± 0.12	0.900 ± 0.11	0.867 ± 0.13
Personal Webs-LSA	0.478 ± 0.07	1.000 ± 0.00	0.983 ± 0.05	0.944 ± 0.10	0.925 ± 0.10	0.907 ± 0.12
	***Strongly related* pairs only**					
	Recall	Prec@1	Prec@2	Prec@3	Prec@4	Prec@5
People Overlap	0.319 ± 0.11	0.370 ± 0.25	0.370 ± 0.20	0.290 ± 0.17	0.265 ± 0.14	0.247 ± 0.15
node2vec	0.447 ± 0.25	0.333 ± 0.31	0.352 ± 0.32	0.333 ± 0.27	0.306 ± 0.19	0.274 ± 0.16
Personal Webs-NP	0.507 ± 0.24	0.519 ± 0.28	0.481 ± 0.21	0.420 ± 0.21	0.398 ± 0.21	0.356 ± 0.20
Personal Webs-LSA	0.522 ± 0.23	0.407 ± 0.26	0.407 ± 0.19	0.383 ± 0.20	0.380 ± 0.21	0.356 ± 0.18

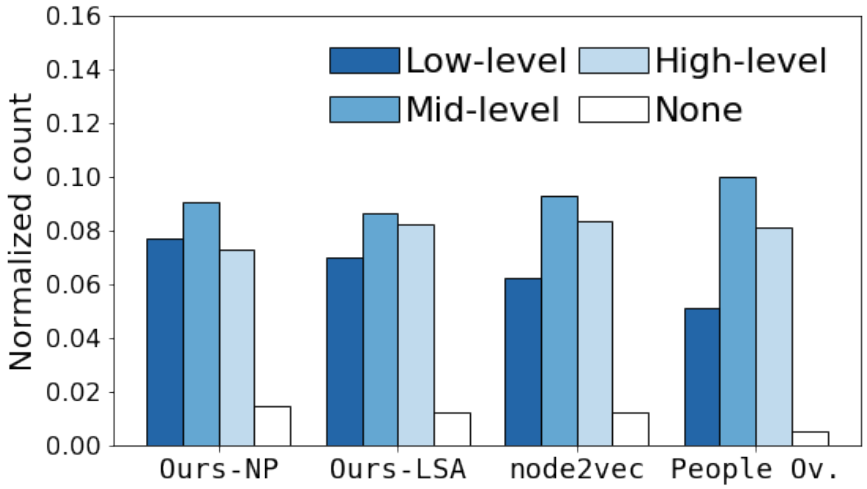


Figure 6.5: Question 1 answers by system across all participants.

Impact of professional roles It should be noted that the standard deviations in Table 6.1 are relatively high. We hypothesize that the variance can be partially explained by participants' varying roles in the workplace, which we found to be correlated with interpretations of the term "activity". For example, our representations performed well for the senior-level participants with many ongoing activities, whereas the People Overlap baseline worked well for junior-level employees with fewer ongoing professional activities. We found that senior-level participants (e.g., principal researchers, managers) tended to have many ongoing activities involving the same group of people, so more fine-grained textual cues were needed to distinguish these activities. By contrast, participants with fewer ongoing professional activities (e.g., interns, individual contributors) tended to

Table 6.2: Average relatedness grade (Question 2) out of 4 across participants P1-P10. Parentheses: Each system's rank per participant; lower is better. Top: All entity pairs; Bottom: *Email-Email* pairs only. Last column: Grades and ranks averaged across all participants.

Participant **# entities in $\mathcal{G}$**	**P1** 157	**P2** 258	**P3** 320	**P4** 303	**P5** 256	**P6** 291	**P7** 203	**P8** 232	**P9** 1468	**P10** 1637	**Avg. grade (rank)**
All pairs of entities											
People Overlap	2.00 (4)	**2.47 (1)**	2.67 (4)	1.87 (4)	**2.77 (1)**	**2.00 (1)**	2.00 (2)	2.00 (3)	2.43 (3)	2.13 (3)	2.22 ± 1.23 (2.60)
node2vec	**2.33 (1)**	2.40 (2)	3.07 (3)	1.93 (3)	2.33 (2)	1.87 (2)	1.80 (3)	1.93 (4)	2.20 (4)	1.73 (4)	2.16 ± 1.38 (2.80)
Personal Webs-NP	2.27 (2)	1.93 (4)	**3.53 (1)**	**2.13 (1)**	2.27 (3)	1.87 (2)	1.80 (3)	**2.53 (1)**	**2.73 (1)**	2.60 (2)	2.37 ± 1.43 (**2.00**)
Personal Webs-LSA	2.13 (3)	2.13 (3)	3.27 (2)	2.07 (2)	2.27 (3)	1.87 (2)	**2.27 (1)**	2.47 (2)	2.53 (2)	**2.80 (1)**	2.38 ± 1.38 (2.10)
***Email-Email* pairs only**											
People Overlap	2.60 (2)	**2.67 (1)**	2.44 (4)	1.75 (3)	2.55 (4)	1.69 (4)	**2.20 (1)**	2.33 (3)	2.46 (2)	2.13 (3)	2.26 ± 1.30 (2.70)
node2vec	2.60 (2)	1.88 (3)	2.78 (3)	1.80 (2)	**3.71 (1)**	**2.00 (1)**	1.00 (3)	1.62 (4)	2.14 (4)	1.73 (4)	2.07 ± 1.39 (2.70)
Personal Webs-NP	2.40 (4)	1.83 (4)	**3.29 (1)**	1.67 (4)	3.62 (2)	**2.00 (1)**	1.00 (3)	**2.57 (1)**	**2.50 (1)**	2.33 (2)	2.40 ± 1.40 (2.30)
Personal Webs-LSA	**2.80 (1)**	2.29 (2)	2.88 (2)	**2.00 (1)**	3.62 (2)	1.89 (3)	2.11 (2)	2.43 (2)	2.42 (3)	**2.79 (1)**	2.54 ± 1.30 (**1.90**)

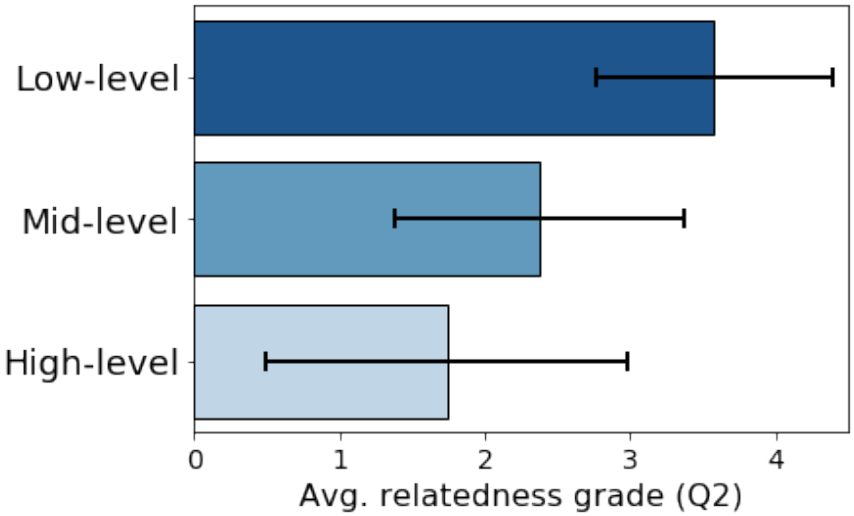


Figure 6.6: Question 2 answer averages stratified by answers to Question 1.

view their activities as shaped primarily by the people involved.

Per-participant performance We explore this effect further in Table 6.2, where we demonstrate each system's performance *per participant* along with the size of each participant's Personal Web $\mathcal{G}$ as measured by the number of entities in the graph. In Table 6.2, the first group of rows gives average grades for Question 2 (out of 4) and average system ranks (lower is better) for all pairs of rated pairs. We find that our NP and LSA representations perform the best overall, whereas the performance of node2vec is middling and People Overlap is more polarized. The second group of rows gives the same information for *Email-Email* pairs only. Here our LSA representations perform by far the best, likely because taking the SVD of the document-term matrix better groups entities into high-level "topics" [Deerwester et al., 1990].

Correlation between Q1 and Q2 Figure 6.5 shows *what kinds of related* items each system found according to participants' responses to Question 1 ("Why do you think the system related this pair of items?"), with answers aggregated across all participants. All methods found a plurality of *mid-level related* pairs (e.g., participating in the same long-term project or activity), but NP found the most pairs with *low-level* (e.g., short term task) relationships. By contrast, People Overlap found the fewest *not related* pairs, which is intuitive as it is a high-precision baseline (Table 6.1, top). We also find that participants' responses to Question 1 correlate with their responses to Question 2 ("In your opinion, how related is this pair of items?"), as demonstrated by Figure 6.6. In particular, across all participants, there appeared to be a strong consensus that short-term tasks corresponded to the strongest relations between entities, with *low-level related* pairs of entities receiving 3.579 ± 0.82 points (out of 4) for Question 2, on average. This correlation suggests that while individuals may have interpreted the notion of "activity" in different ways, their ratings for Questions 1 and 2 were consistent.

Overall, our results demonstrate the strengths of our graph-based representations over several strong baselines. In particular, while there may not be a "one-size-fits-all" approach to activity discovery, we find that our representations perform well for people with many ongoing activities, e.g., senior-level employees. Future work could further investigate how professional roles correlate with individuals' perceptions of, and participation in, activities.

6.6 Task-Based Evaluation

To complement our small-scale rating task, we conduct additional experiments on a large public email dataset to demonstrate the versatility of our unsupervised representations.

6.6.1 Experimental Setup

Data We use the Avocado email dataset, which comprises the inboxes of several hundred employees of a now-defunct technology company referred to as Avocado.[1] We filter each inbox following the processing described in Ch. 6.5.1. From each filtered inbox we extract a graph consisting of *Email*, *Contact*, and *File* entities. We define the *Contact-Email* and *Email-Email* relations the same way as in the previous section, and define an *Email-File* relation that connects emails to their attachments.

Table 6.3 provides aggregate statistics of all Personal Webs. Following our findings in Ch. 6.5.3 on role-based differences of individuals' activities, we stratify our dataset by size, roughly along classes of professional roles (e.g., worker, middle management, officer) [Jin et al., 2019b]: *Small*

[1] https://catalog.ldc.upenn.edu/LDC2015T03

Table 6.3: Aggregates from the Avocado inboxes, stratified by size, after filtering and preprocessing following Ch. 6.5.1. All Personal Web statistics are medians.

	Personal web		**Entity types**		
	# nodes	# edges	# emails	# contacts	# files
Small ($n = 45$)	501	1024	373	39	71
Medium ($n = 76$)	2862	6160	2 414	135	335
Large ($n = 7$)	12 759	37 119	10 755	303	1 457

inboxes have 200–1 000 nodes in their respective graphs, *Medium* have 1 000–10 000, and *Large* have $\geq$10 000.

Task Following prior work in activity modeling [Qadir et al., 2016], we use an email recipient recommendation task, i.e., link prediction among *Email* and *Contact* nodes in the Personal Web, to evaluate the downstream utility of our graph-based representations. Per employee in the Avocado dataset, we construct a test set consisting of the last 8 months of emails from his or her inbox. We retain only the first email per email thread. For each test email, we remove the last recipient on the To line, following the setup in [Qadir et al., 2016]. We discard emails with fewer than two recipients and emails whose last recipient was never seen in the training set, meaning that we evaluate under a transductive assumption (Chapter 2.2.1). For the graph-based methods, we construct a graph given the missing edges between emails and contacts, and learn the respective model over the partial graph.

At test time, given an email with the last recipient missing, each approach creates a ranked list of candidate recipients r', excluding the test email's sender s and observed recipients r. For the methods that represent entities as vectors, we return the candidate recipients r' in order of increasing Euclidean distance from the test email's vector representation. Following [Qadir et al., 2016], our metrics of choice are hits@k for $k \in \{1, 2\}$, which quantifies the fraction of predictions where the correct recipient is ranked in the top k results, and mean reciprocal rank (MRR).

Baselines We compare to the following baselines:

- **Random**: We rank the recipients r' in random order.
- **Frequent Recipients**: We rank recipients r' by $P(r' = u')$ for all people u' observed in the inbox.
- **Conditioned On Sender**: Similar to Frequent Recipients, but conditioned on sender s, e.g. $P(r' = u'|s)$.

Table 6.4: Performance in the recipient recommendation task averaged across all Avocado inboxes. Top performer(s) per metric shaded. ▲: Significant over all methods not marked with † for a two-sided t-test at $p < 0.01$.

	Hits@1	Hits@2	MRR
Random	0.019 ± 0.023	0.038 ± 0.040	0.081 ± 0.060
Freq. Recipients	0.107 ± 0.106	0.184 ± 0.136	0.229 ± 0.105
Cond. On Sender	$0.143 \pm 0.094^{\dagger}$	$0.247 \pm 0.113^{\blacktriangle}$	$0.282 \pm 0.090^{\dagger}$
Average NP	0.128 ± 0.088	0.209 ± 0.119	0.259 ± 0.102
node2vec	0.062 ± 0.072	0.092 ± 0.108	0.126 ± 0.114
Personal Webs-NP, $\lambda = 10^{-1}$	0.111 ± 0.059	0.182 ± 0.096	0.225 ± 0.082
Personal Webs-NP, $\lambda = 10^{0}$	$0.158 \pm 0.084^{\blacktriangle}$	$0.247 \pm 0.105^{\blacktriangle}$	$0.290 \pm 0.089^{\blacktriangle}$
Personal Webs-NP, $\lambda = 10^{2}$	$0.143 \pm 0.085^{\dagger}$	$0.225 \pm 0.112^{\dagger}$	$0.267 \pm 0.093^{\dagger}$
Personal Webs-LSA	0.110 ± 0.093	0.180 ± 0.126	0.224 ± 0.111
Personal Webs-LDA	0.082 ± 0.080	0.141 ± 0.123	0.189 ± 0.111

- **Average NP**: We represent each email as a noun-phrase frequency vector and each candidate recipient u' as the average of all vectors representing emails she has sent or received.
- **node2vec**: We report the best node2vec performer on a grid search over the walk length $l \in \{10, 80\}$, in-out parameter $q \in \{1, 2\}$, and embedding dimension $d \in \{32, 128\}$. All other parameters are set to their defaults.

Personal Webs variants We learn our graph-based representations with the following variants:

- **NP**: We vary $\lambda \in \{10^{-1}, 10^{0}, 10^{2}\}$.
- **LSA**: We set $\lambda = 10^{2}$.
- **LDA**: We decompose the document-term matrix with Latent Dirichlet Allocation (LDA [Blei et al., 2003]) before propagation. We set $\lambda = 10^{2}$ and the number of latent topics to 10.

6.6.2 Results and Discussion

As shown by Table 6.4, our graph-based representations outperform or tie all baselines, suggesting their versatility in activity-centric tasks for which they are not directly optimized. Specifically, our NP representations with $\lambda = 10^{0}$ perform best on average across all Avocado inboxes, tied with the Conditioned On Sender baseline for Hits@1 but otherwise around 1 percentage point or higher than the best baseline. Our NP representations outperform our LSA representations by a significant margin in the recipient recommendation task. As demonstrated in Ch. 6.5.3, our NP

representations are best at identifying strongly-related pairs of entities (Table 6.1, bottom), which we hypothesize may be most useful for email recipient recommendation.

These results also offer insight into the effects of constructing interaction graphs from personal information collections, which is a key design choice of this work. For example, a larger value of the regularization hyperparameter λ translates into more similar entity representations for entities that are closely connected in the graph. In the context of recipient recommendation, this corresponds to people who co-occur often on emails, leading to better prediction performance.

Cross-modal versus single-modality approaches The Personal Web is inherently a cross-modal representation of personal information collections, capturing both interaction structure and text. Our node2vec baseline is also cross-modal, as we learn node representations over a graph connecting people, documents, and noun phrases in those documents. By contrast, our Average NP baseline, which represents each person as an average noun phrase frequency vector, can be seen as a *content-only* version of our NP representation. Likewise, our Frequent Recipients and Conditioned on Sender baselines can be seen as *interaction structure-only*, as they do not consider textual content.

The fact that our representations outperform all baselines suggests that joint modeling of interaction and content is key to improving search and recommendation in personal information collections. However, it should also be noted that the correct inductive biases and model design are required to achieve this good performance: Our node2vec baseline performs worse than our text- and structure-only baselines even though it is cross-modal, suggesting that it is ill-suited to small personal information collections. We hypothesize that its use of random walks may introduce spurious relations between objects and terms in the Personal Web.

6.7 Scalability Evaluation

Finally, we examine how model training scales in both offline and online settings. All experiments were run on a single personal laptop with an Intel i7 1.90GHz processor and 16GB RAM.

6.7.1 Offline Setting

We compare the efficiency of offline inference with our NP and LSA variants versus node2vec. We randomly sample one Avocado inbox of each size (Table 6.3) and train NP, LSA, and node2vec five times over each inbox, reporting the average training time in seconds in Figure 6.7. We find that learning our NP representations is consistently faster than training node2vec (between $3\times$ to $8\times$) since the latter requires computing expensive random walks across the graph in order to generate

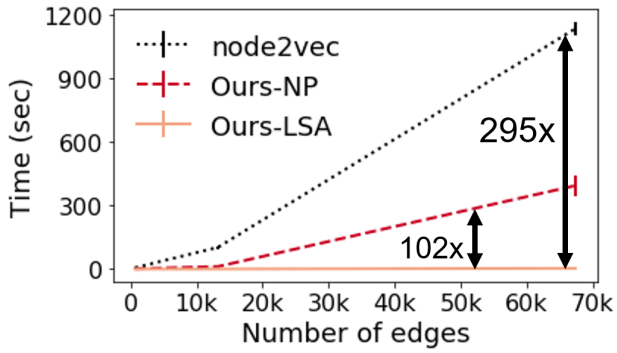


Figure 6.7: Average time in seconds to train our NP and LSA representations offline versus offline node2vec as the graph size increases, measured by the number of edges.

context for each node. Our LSA variant is even more efficient, up to $102\times$ faster than NP, and up to $295\times$ faster than node2vec, running in only a few seconds on a graph of 70k edges. This is because taking the rank-r SVD, which is $O(r^2n)$ for n nodes in the graph [Halko et al., 2011], results in a constant number of (latent) textual attributes r being propagated across the graph. For a graph of m edges, the overall complexity of the LSA variant is $O(r^2n + rm) = O(m)$ for a constant $r \ll n < m$.

6.7.2 Online Setting

We next measure the difference between online and offline model inference. Recall from Ch. 6.4.2.2 that our LSA variant is not eligible for incremental training because the SVD of the document-term matrix must be recomputed from scratch each time new data arrive, so we only experiment with the NP representations here. Figure 6.8 reports the average number of seconds per edge processed, and the average efficiency gain, on all Avocado corpora stratified by size (as detailed earlier in Table 6.3), for our offline- and online-learned NP representations. Online NP, which updates representations incrementally per new edge, is up to $470\times$ faster than offline NP, taking on average less than 1 second per edge on the *Large* corpora.

Figure 6.9 demonstrates how the error of a single, *offline*-learned set of NP representations increases as new edges arrive over time for a randomly sampled *Medium* Avocado inbox. Here we measure the difference between the batch-learned offline representations and the most current, online-learned version with mean squared error. Evidently the error increases approximately linearly as edges arrive, suggesting that for individuals with a high volume of incoming data (e.g., upper-level roles like managers, executives), updating the representations of their personal information objects in an online manner becomes more important.

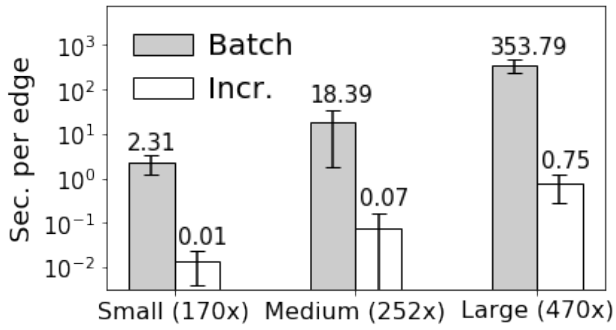


Figure 6.8: Log-scale training time for online and offline NP in seconds per edge, averaged across all inboxes of each size. Average efficiency gain in parentheses.

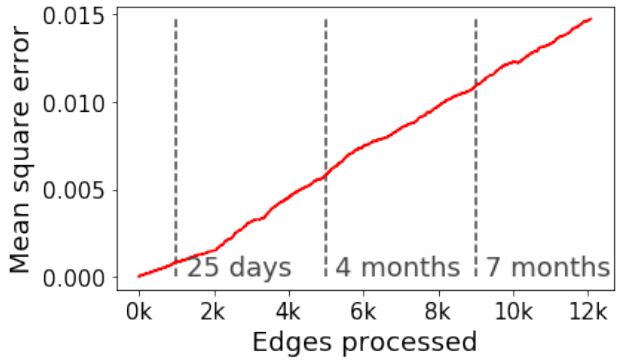


Figure 6.9: Mean square error of a set of static NP representations learned once and not updated as new edges arrive over the course of a year for a *Medium* Avocado graph.

6.8 Conclusion

In this chapter, we studied the task of learning higher-level activity structure over interactions and content in personal information collections, in order to help people manage their information more effectively. We introduced the concept of the Personal Web, an integrated structure and content view of personal information collections, and proposed an efficient joint graph propagation-based approach to learn and update entity representations in the Personal Web. In keeping with the generality of our goal and the versatility of our unsupervised representations, we conducted multiple evaluation tasks. In this first, we uniquely collected direct relevance feedback from a small set of participants by running our own model, eyes-off, over participants' personal information. In the second, we devised a downstream email recipient recommendation task over a public email dataset. In all experiments, we demonstrated that our representations capture personal notions of activity-based relatedness while maintaining efficiency in an online setting.

CHAPTER 7

Classifying Documents with Cross-Modal Inputs

The material in this chapter is derived from the paper "Late-Stage Fusion: Revisiting Node Classification with a Practical and Effective Baseline", which is under submission at the Thirty-ninth International Conference on Machine Learning (ICML), 2022.

7.1 Introduction

The previous chapters of this thesis considered the fundamental graph learning task of link prediction, or inferring unseen edges between pairs of nodes in a graph. In this chapter, we turn to another central graph learning task, that of node classification, which we introduced in Chapter 2.2.2. Node classification refers to the task of classifying examples in a dataset for which interactions or relationships between examples are known [Neville and Jensen, 2007]. For example, in text mining and analysis, it is often of interest to classify documents by topic or category. When relationships between documents are available, for example hyperlinks between Web pages or citations between scientific articles, the task becomes that of node classification: Given a graph consisting of a set of nodes with textual features and classes, as well as graph links or edges between the nodes, classify the nodes with unknown labels [Kipf and Welling, 2017, Hu et al., 2020].

In recent years, successful approaches to node classification have been dominated by deep graph convolutional networks or GCNs [Kipf and Welling, 2017, Veličković et al., 2018], which, as described in Chapter 2.4, use neural "message passing" (MP) to jointly encode the graph's interaction structure and content features in an end-to-end trainable architecture [Hamilton, 2020]. Under MP, each node's hidden representation is computed as a combination of its own representation and the aggregated representations of its neighbors in the graph. For all of its successes, however, neural MP has a strong inductive bias toward **homophilous** graphs in which connected pairs of nodes in the graph are likely to belong to the same class. It therefore underperforms on **heterophilous** graphs in which nodes of different classes are more likely to be connected than nodes of the same class, mostly due to its tendency to *oversmooth* [Chen et al., 2020b, Klicpera et al., 2019, Pei et al.,

Table 7.1: Feature-only models outperform GCNs under heterophily, whereas GCNs outperform feature-only models under homophily. Our simple ensembling approach called Late-Stage Fusion (LSF) achieves robust performance independent of the homophily level.

	Cornell (Heterophilous)	Arxiv (Homophilous)
Best feature-only model	84.86	69.93
Best GCN model	77.84	71.74
LSF	**86.22**	**74.57**

2020, Oono and Suzuki, 2020, Yan et al., 2021]. In heterophilous settings, simple non-relational models like MLP that ignore the graph structure and use only the node features are known to be competitive, often outperforming GCNs [Zhu et al., 2020]. Table 7.1 shows how performance of feature-only and GCN-based models can vary with different homophily levels. Even the best GCN model still has a 7% absolute drop in accuracy on the heterophilous Cornell graph when compared to simple feature-only models.

While recent work has proposed to adjust GCNs for heterophilous settings [Zhu et al., 2020, 2021, Chien et al., 2021, Suresh et al., 2021], such methods typically focus on heuristic transformations of the data or GCN architecture to better match the inductive bias of neural MP. Moreover, while homophily is typically reported as a global statistic, homophily has been found to vary *locally* in most network datasets, as shown in Figure 7.1 [Suresh et al., 2021, Lim et al., 2021, Yan et al., 2021]. This motivates the need for practical, robust methods that are flexible enough to account for varying degrees of homophily, both *within* and *across* graphs.

7.1.1 Contributions

Toward this goal, in this work we investigate a simple, yet surprisingly effective ensembling baseline for node classification called **Late-Stage Fusion** (**LSF**). Focusing on graphs with textual features, our ensembling approach combines the predictions of base GCN models with the predictions of base *text-only* models (e.g., MLP for vector features, deep language models for raw textual content) via a weighted majority vote. Departing from classical ensembling, which combines the same type of model trained on different views of the data to reduce variance [Breiman, 1996, Bartlett et al., 1998, Kuncheva and Whitaker, 2003], LSF leverages the key insight that feature-only and relational models perform differently, and often *complementarily*, at varying levels of homophily. We provide a theoretical analysis of LSF that shows how it can provably improve the ensembling error of the weaker of the two model groups, leading to robust performance regardless of homophily

levels. In our extensive empirical evaluation, we compare to 13 competitive GCN baselines across 8 diverse graph datasets and show that LSF achieves equivalent or better performance in all scenarios. Given the simplicity and effectiveness of LSF, we believe that it should serve as a baseline and starting point for future model development in node classification.

Our contributions are as follows:

- **New model fusion approach** (Ch. 7.4): We propose Late-Stage Fusion (LSF), a simple but powerful cross-modal model fusion approach for node classification. LSF combines the classification outputs of graph learning models with those of text learning models.
- **Theoretical characterization** (Ch. 7.4.2): We provide a theoretical characterization that implies LSF's strong performance regardless of the homophily level of the graph. Considering one group of graph learning models and one group of text learning models, we prove that LSF improves the ensembling error of the weaker of the two model groups, leading to robust performance regardless of homophily levels.
- **Extensive evaluation** (Ch. 7.5): We compare LSF to 13 competitive GCNs across eight diverse graph datasets with varying homophily patteerns. LSF meets or exceeds state-of-the-art performance on all datasets. Despite its simplicity, LSF on average reduces error by 4.9% compared to the best-performing baseline on each dataset. In contrast, GraphSAGE, the baseline with most robust performance as homophily varies, on average increases error by 11.8%. In further analysis, we demonstrate how the superior performance of LSF mainly comes from its successful modeling of *local* homophily patterns in a graph, allowing the feature information to dominate in low-homophily regions, and leveraging the feature and relation correlations in higher-homophily regions.

7.2 Related Work

Variants of the graph convolutional network or GCN [Kipf and Welling, 2017] have enjoyed immense success across a variety of graph-based machine learning tasks and application domains [Wu et al., 2020b], the most common of which is node classification [Hamilton et al., 2017, Veličković et al., 2018, Wu et al., 2019, Hamilton, 2020]. However, the strong relational inductive bias of neural message passing (MP) is known to lead to lower performance on heterophilous graphs [Xu et al., 2019, Zhu et al., 2020, 2021], and also has strong connections to the oversmoothing problem [Oono and Suzuki, 2020, Liu et al., 2020a, Yan et al., 2021]. In this section, we discuss various directions toward addressing these drawbacks.

Notions of homophily Similar to our goal of modeling local homophily, various studies in network science have proposed to measure homophily patterns in graphs with group-level [Currarini et al., 2009] and node-level [Interian and Ribeiro, 2018] statistics. More recently in the graph neural networks community, recent work has argued that graph-level homophily statistics may not fully characterize how the class labels correlate with the edges. Suresh et al. [2021] analyze how homophily patterns can be distributed unequally across two graphs with the same global level of homophily. Lim et al. [2021] argue that global summary statistics for homophily may be misleading because they do not account for number of classes and imbalance in class distributions.

Handling heterophily Several heuristics for connecting distant or non-neighboring nodes in a graph have been proposed to improve MP under heterophily. One strategy introduces skip connections among different layers, such that hidden features from earlier layers, which are less influenced by the graph structure, can provide input into the final class predictions [Hamilton et al., 2017, Xu et al., 2018, Chen et al., 2020b, Zhu et al., 2020]. Other approaches modify the input graph's topology in order to introduce new dependencies between nodes and relax the relational inductive bias imposed by the original graph [Pei et al., 2020, Suresh et al., 2021, Yan et al., 2021]. Note, however, that all of these methods are all still MP architectures at their core, and make class predictions according to (some) graph structure, whereas LSF is more generalized because it allows feature-only learners to override the predictions of relational learners.

Disentangling features and relations A related direction of research seeks to disentangle the node feature information from the graph's relational structure. Variants of the "predict-then-propagate" framework originally introduced by Klicpera et al. [2019] have been proposed, both end-to-end trainable [Liu et al., 2020a] and modular two-step [Huang et al., 2021]. While these methods were not originally designed to handle heterophily, more recent approaches tailored toward separating features and relations have been proposed specifically in the context of heterophilous graphs [Chien et al., 2021, Ma et al., 2021b, Lim et al., 2021]. As these are the closest to LSF, we compare to several competitive variants in our experiments.

7.3 Preliminary Analysis

We consider the transductive node classification task as introduced in Chapter 2.2.2 over a singly-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Given a training set consisting of nodes $\mathcal{V}_{\text{train}} \subset \mathcal{V}$ and their labels, all node features $\mathbf{X}$, and the adjacency matrix $\mathbf{A}$, predict the labels of the remaining nodes $\mathcal{V}_{\text{test}} = \mathcal{V} \backslash \mathcal{V}_{\text{train}}$. In practice, we only consider textual node features in this chapter. However, for generality we will not limit our discussion in this section to text, since nodes may be associated with real-

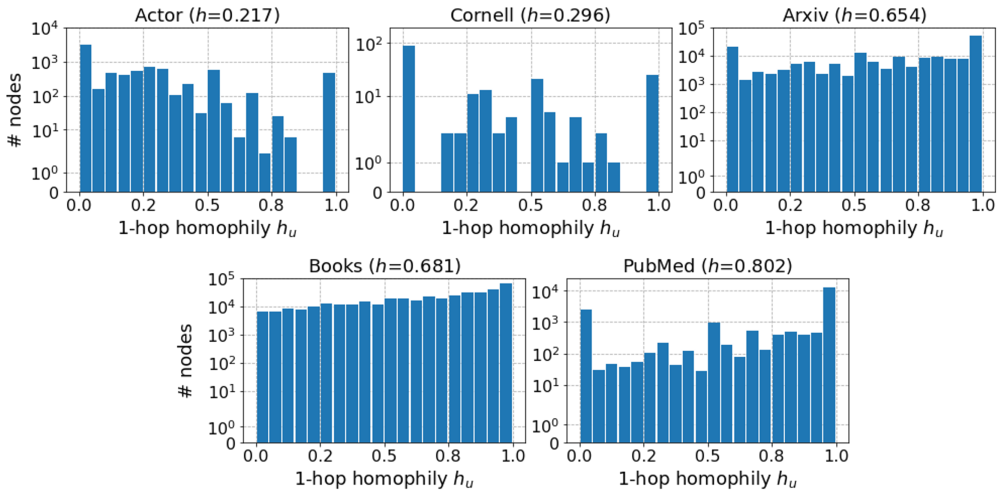


Figure 7.1: 1-hop homophily (Equation 7.2) characterizes each graph's class distribution on a local level: For example, even though PubMed has a higher *global* homophily score h (Equation 7.1) than Books, PubMed has a different distribution of *local* 1-hop homophily scores.

valued or categorical features beyond text, and our proposed approach LSF can handle nodes of any feaure type.

7.3.1 Global and Local Homophily

The homophily of a network measures the degree to which similar nodes tend to interact [McPherson et al., 2001]. In the context of node classification, we consider similarity purely from the perspective of node classes, with higher-homophily graphs connecting more pairs of nodes of the same class than across different classes. This can be quantified at various levels of granularity in a graph, from a per-node level [Interian and Ribeiro, 2018] to a per-group [Currarini et al., 2009] or whole-graph level [Zhu et al., 2020].

In this chapter, we will primarily consider whole-graph global homophily and per-node local homophily. Toward the former, Zhu et al. [2020] propose to measure a graph's **global homophily** as the fraction of edges connecting nodes of the same class, divided by the total number of edges in the graph:

$$h = \frac{\{(u,v)\ \forall (u,v) \in \mathcal{E} \mid y_u = y_v\}}{|\mathcal{E}|}. \tag{7.1}$$

While this statistic is useful for succinctly summarizing a graph, it may not capture finer-grained variations in the graph's class distribution. Toward analyzing homophily on a *local* level, we consider a simple node-level statistic which we call **1-hop homophily** Given a node u with graph

neighbors $\mathcal{N}_u$, we define u's 1-hop homophily score h_u as

$$h_u = \frac{|\{y_v = y_u \mid \forall v \in \mathcal{N}_u\}|}{|\mathcal{N}_u|}, \tag{7.2}$$

that is, the proportion of u's neighbors belonging to the same class as u.

The 1-hop homophily statistic has been explored previously in the network science literature for measuring network polarization [Interian and Ribeiro, 2018]. However, in the context of analyzing the performance of graph neural networks, it has primarily been considered at a *global* level [Pei et al., 2020] by taking the average h_u across all nodes $u \in \mathcal{V}$, whereas we use it as a tool for analysis on a local, per-node level.

Figure 7.1 provides examples of graphs in terms of both global homophily h (Equation 7.1) and the distribution of their local homophily scores h_u in bins. We observe that two graphs of similar global homophily score (Arxiv, Books) may have very different distributions of local homophily. Moreover, the graphs with low global homophily (Actor, Cornell) still have a relatively large proportion of nodes with high 1-hop homophily scores, and vice-versa.

7.4 Methodology

7.4.1 Late-Stage Fusion Overview

Our preliminary analysis motivates us to develop an approach flexible enough to account for varying degrees of homophily both *within* and *across* graphs. Toward this goal, we investigate a simple node classification baseline that we call **Late-Stage Fusion** (**LSF**). For a given node u, we define the LSF classification rule for u as follows:

$$\hat{y}_u = \arg\max_{c \in \mathcal{C}} \left\{ \sum_{i=1}^{N} w_i \mathbb{I}[\hat{y}_{i,u}^{\text{TEXT}} = c] + \sum_{j=1}^{M} w_j \mathbb{I}[\hat{y}_{j,u}^{\text{REL}} = c] \right\}.$$

The first term refers to a weighted sum over the predictions of $N \geq 1$ feature-only base models f_i^{TEXT} (e.g., MLP), where each $\hat{y}_{i,u}^{\text{TEXT}}$ is defined as follows:

$$\hat{y}_{i,u}^{\text{TEXT}} = f_i^{\text{TEXT}}(\mathbf{x}_u; y_u), \;\; i = 1 \ldots N.$$

Likewise, the second term refers to a weighted sum over $M \geq 1$ relational base models f_j^{REL} (e.g., GCN), where each $\hat{y}_{j,u}^{\text{REL}}$ is defined as follows:

$$\hat{y}_{j,u}^{\text{REL}} = f_j^{\text{REL}}(\mathbf{A}, \mathbf{x}_u; y_u), \;\; j = 1 \ldots M.$$

The weights w_i and w_j are used to weight each base classifier's predictions. These weights may be uniform, treated as hyperparameters, or even learned on a separate hold-out set. In practice, for simplicity, we search over a single ratio hyperparameter $\alpha \in [0, 1]$ on the validation set such that $w_i = \frac{\alpha}{N}$ and $w_j = \frac{1-\alpha}{M}$.

7.4.2 Theoretical Characterization

In this section, we characterize properties of LSF theoretically. We show that, for an established definition of voting ensemble error from the classical ensembling literature [Bartlett et al., 1998], ensembling across feature-only and relational models is guaranteed to improve upon the error of the weaker group of models (whose performance differs under varying homophily) for any given node.

Assume we are given a data point u with label $y_u = c$ and an ensemble of L base learners $f_i(u)$. The **margin** [Bartlett et al., 1998] of the ensemble is defined as the weighted proportion of base learners $f_i(u)$ that classify u correctly (i.e., predict class c), less the maximum weighted proportion of base learners that vote for the *same* incorrect class $c' \neq c$.

Formally, the margin, assuming equal weights on all L base learners $f_i(u)$, is written as

$$\text{Margin}_u = \frac{1}{L}\left(\sum_{i=1}^{L} \mathbb{I}[f_i(u) = c] - \max_{c' \neq c} \sum_{i=1}^{L} \mathbb{I}[f_i(u) = c']\right) \in [-1, 1]. \tag{7.3}$$

The goal of the voting ensemble is to maximize this quantity. In particular,

$$\text{accuracy}_u = \begin{cases} 1, & \text{if Margin}_u > 0 \\ 0, & \text{otherwise} \end{cases}.$$

Thus, the margin may be intuited as a measure of confidence for the voting ensemble, with a greater absolute value indicating a more confident ensemble.

7.4.2.1 Uniform Weights

We start with a simplified version of LSF in which all weights w_i and w_j are uniform. Given the definition of margin in Equation 7.3, we have the following:

Lemma 7.4.1. *Assume N base learners f_i^{TEXT} and M base learners f_j^{REL}. For a given node u, if*

we weight each base learner with a uniform ratio of $w_i = w_j = \frac{1}{N+M}$*, then*

$$\begin{aligned}
&\text{Margin}_u^{\text{LSF}} > \min(\text{Margin}_u^{\text{TEXT}}, \text{Margin}_u^{\text{REL}}), \\
\iff &(\text{Margin}_u^{\text{LSF}} > \text{Margin}_u^{\text{TEXT}}) \vee (\text{Margin}_u^{\text{LSF}} > \text{Margin}_u^{\text{REL}}),
\end{aligned}$$

where $\text{Margin}_u^{\text{LSF}}$ *refers to the margin of LSF over the ensemble of* $N + M$ *base learners,* $\text{Margin}_u^{\text{TEXT}}$ *refers to the margin of the* N *feature-only learners, and* $\text{Margin}_u^{\text{REL}}$ *refers to the margin of the* M *relational learners.*

Proof. Given node u with label y_u, let us assume that $\text{Margin}_u^{\text{TEXT}} > \text{Margin}_u^{\text{REL}}$, i.e.,

$$\begin{aligned}
&\frac{1}{N}\left(\sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c - \max_{c_1' \neq c} \sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c_1'\right) \\
&> \frac{1}{M}\left(\sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c - \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'\right).
\end{aligned} \tag{7.4}$$

Note that we use c_1' and c_2' to indicate that the maximally weighted incorrect class prediction may differ between the feature-only learners and the relational learners.

First, notice that

$$\begin{aligned}
\text{Margin}_u^{\text{LSF}} &= \frac{1}{N+M}\left[\sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c + \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c \right. \\
&\quad \left. - \max_{c' \neq c}\left(\sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c' + \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c'\right)\right] \\
&\geq \frac{1}{N+M}\left(\sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c + \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c \right. \\
&\quad \left. - \max_{c_1' \neq c} \sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c_1' - \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'\right)
\end{aligned}$$

because for any two functions f and g, $\max(f + g) \leq \max(f) + \max(g)$; in this case,

$$\max_{c' \neq c}\left(\sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c' + \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c'\right)$$

is equal to

$$\max_{c_1' \neq c} \sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c_1' + \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'$$

only when $c_1' = c_2'$, and is otherwise less.

Next, we show that

$$\begin{aligned}
&\frac{1}{N+M}\left(\sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c + \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c\right. \\
&\left. - \max_{c_1' \neq c} \sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c_1' - \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'\right) \\
&> \text{Margin}_u^{\text{REL}}.
\end{aligned}$$

We cross-multiply the inequality in Equation 7.4 by the leading scalars and add

$$M\left(\sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c - \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'\right)$$

to both sides of the inequality, yielding

$$\begin{aligned}
&M\left(\sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c - \max_{c_1' \neq c} \sum_{i=1}^{N} \mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c_1'\right. \\
&\left. + \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c - \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'\right) \\
&> N\left(\sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c - \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'\right) \\
&+ M\left(\sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c - \max_{c_2' \neq c} \sum_{j=1}^{M} \mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2'\right),
\end{aligned}$$

which can be rewritten as

$$\begin{aligned}
&\frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}}=c+\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}}=c\right.\\
&\left.-\max_{c_1'\neq c}\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}}=c_1'-\max_{c_2'\neq c}\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}}=c_2'\right)\\
&>\frac{1}{M}\left(\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}}=c-\max_{c_2'\neq c}\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}}=c_2'\right)\\
&=\text{Margin}_u^{\text{REL}}.
\end{aligned}$$

Thus, we have that

$$\begin{aligned}
\text{Margin}_u^{\text{LSF}}&\geq\frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}}=c+\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}}=c\right.\\
&\quad\left.-\max_{c_1'\neq c}\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}}=c_1'-\max_{c_2'\neq c}\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}}=c_2'\right)\\
&>\text{Margin}_u^{\text{REL}}\\
&=\min(\text{Margin}_u^{\text{TEXT}},\text{Margin}_u^{\text{REL}}).
\end{aligned}$$

The same reasoning can be used to show that when $\text{Margin}_u^{\text{REL}} > \text{Margin}_u^{\text{TEXT}}$, then $\text{Margin}_u^{\text{LSF}} > \text{Margin}_u^{\text{TEXT}} = \min(\text{Margin}_u^{\text{TEXT}}, \text{Margin}_u^{\text{REL}})$, which completes the proof. □

Since Lemma 7.4.1 holds for every u, on a global level it means that the accuracy score of LSF will be no worse than the accuracy score of the weaker of the two model groups.

Lemma 7.4.2. *Let $\mathcal{M}^{\uparrow}$ refer to the stronger of the two base learner sets $\{f_i^{\text{TEXT}}\}$ and $\{f_j^{\text{REL}}\}$, and let $\mathcal{M}^{\downarrow}$ refer to the weaker of the two base learner sets, such that*

$$\text{Margin}_u^{\mathcal{M}^{\uparrow}} > \text{Margin}_u^{\mathcal{M}^{\downarrow}}.$$

If $\mathcal{M}^{\uparrow}$ and $\mathcal{M}^{\downarrow}$ make different predictions for node u (i.e., $\hat{y}_u^{\mathcal{M}^{\uparrow}} \neq \hat{y}_u^{\mathcal{M}^{\downarrow}}$), and $\text{Margin}_u^{\mathcal{M}^{\uparrow}} > |\text{Margin}_u^{\mathcal{M}^{\downarrow}}|$, then LSF will make the same prediction for node u as $\mathcal{M}^{\uparrow}$, i.e.,

$$\hat{y}_u^{\text{LSF}} = \hat{y}_u^{\mathcal{M}^{\uparrow}}.$$

Proof. If $\text{Margin}_u^{\mathcal{M}^{\uparrow}} > 0$ and $\text{Margin}_u^{\mathcal{M}^{\downarrow}} < 0$, then the prediction of the weaker base learner set will be incorrect, but the prediction of the stronger base learner set will be correct. We know from

Lemma 7.4.1 that $\text{Margin}_u^{\text{LSF}}$ will be no less than that of the weaker base learner set. If in addition, $\text{Margin}_u^{\mathcal{M}^\uparrow} > |\text{Margin}_u^{\mathcal{M}^\downarrow}|$, then

$$\text{Margin}_u^{\text{LSF}} > \text{Margin}_u^{\mathcal{M}^\uparrow} + \text{Margin}_u^{\mathcal{M}^\downarrow} > 0.$$

Then LSF will also make the same prediction as the stronger base learner set. □

Whereas Lemma 7.4.1 says LSF will do no worse than the weaker model group, Lemma 7.4.2 shows how in practice LSF can be as accurate as the stronger model group. If the two types of models (TEXT and REL) are confident on the examples they classify correctly, which often corresponds to heterophily and homophily respectively, and less confident on the examples they classify incorrectly, then in practice the assumption of Lemma 7.4.2 will hold. This means that the accuracy of LSF will be closer to that of the stronger base learners, *in both homophilous and heterophilous regions of the graph*.

7.4.2.2 Non-Uniform Weights

We next extend Lemma 7.4.1 to the case where the feature-only learners are weighted as $w_i = \frac{\alpha}{N}$ and the relational learners are weighted as $w_j = \frac{1-\alpha}{M}$, as per our practical implementation of LSF:

Lemma 7.4.3. *Assume N base learners f_i^{TEXT} and M base learners f_j^{REL} with weighting $w_i = \frac{\alpha}{N}$ and $w_j = \frac{1-\alpha}{M}$, respectively, and $0 < \alpha < 1$. For any given node u, if at least one of the two groups of base learners makes the correct majority vote, i.e.,*

$$\text{Margin}_u^{\text{TEXT}} > 0 \ \vee \ \text{Margin}_u^{\text{REL}} > 0,$$

then

$$\begin{aligned} \text{Margin}_u^{\text{LSF}} &> \min\left(\alpha \cdot \text{Margin}_u^{\text{TEXT}}, (1-\alpha) \cdot \text{Margin}_u^{\text{REL}}\right) \\ \iff &(\text{Margin}_u^{\text{LSF}} > \alpha \cdot \text{Margin}_u^{\text{TEXT}}) \\ &\vee (\text{Margin}_u^{\text{LSF}} > (1-\alpha) \cdot \text{Margin}_u^{\text{REL}}). \end{aligned}$$

Proof. We have that

$$\begin{aligned}
\text{Margin}_u^{\text{LSF}} &= \frac{\alpha}{N}\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c + \frac{1-\alpha}{M}\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c \\
&\quad - \max_{c'\neq c}\left(\frac{\alpha}{N}\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c' + \frac{1-\alpha}{M}\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c'\right) \\
&\geq \frac{\alpha}{N}\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c + \frac{1-\alpha}{M}\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c - \frac{\alpha}{N}\max_{c_1'\neq c}\sum_{i=1}^{N}\mathbb{1}\hat{y}_{i,u}^{\text{TEXT}} = c_1' \\
&\quad - \frac{1-\alpha}{M}\max_{c_2'\neq c}\sum_{j=1}^{M}\mathbb{1}\hat{y}_{j,u}^{\text{REL}} = c_2' \\
&= \alpha\cdot\text{Margin}_u^{\text{TEXT}} + (1-\alpha)\cdot\text{Margin}_u^{\text{REL}}.
\end{aligned}$$

In the case that $\alpha\cdot\text{Margin}_u^{\text{TEXT}} > (1-\alpha)\cdot\text{Margin}_u^{\text{REL}}$ and $\text{Margin}_u^{\text{TEXT}} > 0$, then using the above inequality we have

$$\begin{aligned}
\text{Margin}_u^{\text{LSF}} &\geq \alpha\cdot\text{Margin}_u^{\text{TEXT}} + (1-\alpha)\cdot\text{Margin}_u^{\text{REL}} \\
&> (1-\alpha)\cdot\text{Margin}_u^{\text{REL}} \\
&= \min(\alpha\cdot\text{Margin}_u^{\text{TEXT}}, (1-\alpha)\cdot\text{Margin}_u^{\text{REL}}).
\end{aligned}$$

The same reasoning can be used to show that when $\text{Margin}_u^{\text{REL}} > \text{Margin}_u^{\text{TEXT}}$ and $\text{Margin}_u^{\text{REL}} > 0$, then $\text{Margin}_u\text{LSF} > \min(\alpha\cdot\text{Margin}_u^{\text{TEXT}}, (1-\alpha)\cdot\text{Margin}_u^{\text{REL}})$, which completes the proof. □

Lemma 7.4.3 shows that, for all nodes that at least one of our base model groups correctly classifies, the accuracy of LSF will be no worse than the accuracy of the weaker model group.

Lemma 7.4.4. *Let $\mathcal{M}^{\uparrow}$ refer to the stronger of the two base learner sets $\{f_i^{\text{TEXT}}\}$ and $\{f_j^{\text{REL}}\}$, and let $\mathcal{M}^{\downarrow}$ refer to the weaker of the two base learner sets, such that*

$$\text{Margin}_u^{\mathcal{M}^{\uparrow}} > \text{Margin}_u^{\mathcal{M}^{\downarrow}}.$$

Let $w^{\uparrow}\in\{\alpha, 1-\alpha\}$ refer to the weight of $\mathcal{M}^{\uparrow}$, and $w^{\downarrow}\in\{\alpha, 1-\alpha\}$ refer to the weight of $\mathcal{M}^{\downarrow}$. If $\text{Margin}_u^{\mathcal{M}^{\uparrow}} > 0$ and $w^{\uparrow}\cdot\text{Margin}_u^{\mathcal{M}^{\uparrow}} > |w^{\downarrow}\cdot\text{Margin}_u^{\mathcal{M}^{\downarrow}}|$, then LSF will make the same prediction for node u as $\mathcal{M}^{\uparrow}$, i.e.,

$$\hat{y}_u^{\text{LSF}} = \hat{y}_u^{\mathcal{M}^{\uparrow}}.$$

Proof. If $w^{\uparrow}\cdot\text{Margin}_u^{\mathcal{M}^{\uparrow}} > 0$ and $w^{\downarrow}\cdot\text{Margin}_u^{\mathcal{M}^{\downarrow}} < 0$, then the prediction of the weaker base learner set will be incorrect, but the prediction of the stronger base learner set will be correct. We

Table 7.2: Datasets used in our experiments. The variable h refers to the global homophily score (Equation 7.1).

	# nodes	# edges	# classes	h
Cornell	183	280	5	0.296
Texas	183	295	5	0.061
Wisconsin	251	466	5	0.178
Actor	7,600	26,752	5	0.217
Wiki-CS	11,701	216,123	10	0.654
PubMed	19,717	44,327	3	0.802
Arxiv	169,343	1,157,799	40	0.654
Books	383,583	4,856,894	10	0.681

know from Lemma 7.4.3 that the margin of LSF will be greater than the weighted margin of the weaker base learner set. If in addition, $w^{\uparrow} \cdot \text{Margin}_u^{\mathcal{M}^{\uparrow}} > |w^{\downarrow} \cdot \text{Margin}_u^{\mathcal{M}^{\downarrow}}|$, then:

$$\text{Margin}_u^{\text{LSF}} > w^{\uparrow} \cdot \text{Margin}_u^{\mathcal{M}^{\uparrow}} + w^{\downarrow} \cdot \text{Margin}_u^{\mathcal{M}^{\downarrow}} > 0.$$

Then LSF will also make the same prediction as the stronger base learner set. □

Similar to Lemma 7.4.2, this shows that the weighted ensemble can make predictions in practice that are as accurate as the stronger of the two base learner sets. In the following sections, we will provide empirical analysis, demonstrating that these implications yield substantial performance improvements in practice.

7.5 Evaluation

In our experiments, we consider a large range of graph datasets across the spectrum of homophily, and compare LSF to a variety of graph learning architectures. We demonstrate LSF's consistent improvements regardless of the homophily level of the graph, and provide empirical analyses to further characterize LSF on a local level.

7.5.1 Experimental Setup

Data We consider eight diverse graph datasets. Table 7.2 provides the dataset statistics. From the literature on graph neural networks for heterophily, we use the **Cornell**, **Texas**, **Wisconsin**, and **Actor** graphs from Pei et al. [2020]. The first three graphs represent hyperlinks between webpages;

the latter represents co-occurrence of mentions of actors on Wikipedia. For all graphs, we use the bag-of-words textual node features and random splits provided by the authors.

From the higher-homophily graph datasets more common in the GCN literature, we use the **Wiki-CS** graph of hyperlinks between Wikipedia pages of computer science fields [Mernyei and Cangea, 2020], and the PubMed [Namata et al., 2012] and Arxiv [Hu et al., 2020] citation networks of scientific articles. We also consider **Books**, a large homophilous graph constructed from the GoodReads dump provided by [Wan and McAuley, 2018, Wan et al., 2019], where nodes are books, edges represent pairs of books that are commonly liked by the same users, and class labels correspond to book genres.

For Wiki-CS and Arxiv, we use the splits and dense text representation node features provided by the authors. For PubMed, we started with the original graph of PubMed articles, citation links, and article classes introduced by Namata et al. [2012]. We retrieved the abstracts of all nodes using their PubMed ID from the PubMed 2021 annual baseline.[1] We then split the data into training, validation, and testing nodes. Note that the original GCN paper uses the random splits introduced by Yang et al. [2016] on PubMed. However, we could not use these splits they do not disambiguate the training/test nodes with the original PubMed IDs, and therefore we could not link them to our retrieved textual abstracts. We therefore split the data according to each article's year of publication, which we also retrieved from the PubMed database. Our design choice was motivated by recent guidance from the Open Graph Benchmark [Hu et al., 2020], which suggests that time-based splits test generalization better than random splits. We train up to 2004, validate up to 2007, and test on the remaining data, yielding a 72/19/9 split ratio.

The Books graph is constructed from the raw dump of GoodReads provided by Wan and McAuley [2018], Wan et al. [2019]. As far as we are aware, even though the dump provides a rich and well-annotated graph from the recommender systems domain, it has not been used in the context of node classification before. In our constructed graph, each node is a book from GoodReads, each edge connects books that are frequently liked by the same users, and each class is one of ten book genres. To construct feature vectors for each node as input to MLPs and GCNs, we extract 1024-dimension bag-of-words feature vectors for each node. We also split the nodes by time, which in this context corresponds to each book's year of publication. We train up to 2013, validate up to 2015, and test on the remaining data, yielding a 70/19/11 split ratio.

Baselines We consider the following baselines, grouped by type: (**1**) **Feature-only**: We use MLP as a feature-only baseline. We also consider the BERT language model [Devlin et al., 2019], the basics of which we covered in Chapter 2.5.2, on the PubMed, Arxiv, and Books graphs only, as we were able to retrieve the raw textual snippets associated with the nodes in these graphs;

[1] https://ftp.ncbi.nlm.nih.gov/pubmed/baseline/

(**2**) **Relations-only**: We consider the classic label propagation algorithm [Zhu and Ghahramani, 2002]; and (**3**) **Standard GCNs**: GCN [Kipf and Welling, 2017], GAT [Veličković et al., 2018], GraphSAGE [Hamilton et al., 2017], and SGC [Wu et al., 2019].

We also consider two separate sets of "specialized" GCNs, one for the heterophilous graphs and one for the homophilous graphs: (**1**) **GCNs for heterophily**: On the heterophilous graphs only, we consider Geom-GCN [Pei et al., 2020], GCNII [Chen et al., 2020b], H2-GCN [Zhu et al., 2020], WRGAT [Suresh et al., 2021], GPR-GNN [Chien et al., 2021], MLP+GCN feature pooling [Ma et al., 2021b], and LINKX [Lim et al., 2021]; and (**2**) **GCNs that separate features and relations**: On the homophilous graphs only, we consider APPNP [Klicpera et al., 2019], DAGNN [Liu et al., 2020a], and C&S [Huang et al., 2021].

LSF variants We implement LSF with a simple approach that reuses our baseline models. As the feature-only learners, we take the top-10 best feature-only baseline models for a given dataset in terms of validation accuracy, and treat $N \in 1 \ldots 10$ as a hyperparameter to be selected in tuning LSF. We do the same with the relational learners using our standard GCN baselines in order to set M. We also tune $\alpha \in [0.1, 0.9]$ on the validation set. Section 7.5.2 provides more insight into hyperparameter selection for LSF.

Software For a fair comparison, we implement all models ourselves unless otherwise noted, and select hyperparameters with the Ax model selection framework.[2] We perform model selection over 10 trials, selecting the hyperparameters for the first 5 trials by randomly sampling from a grid, and selecting hyperparameters for the remaining 5 trials using Ax's built-in Bayesian model selection API. Software is written in PyTorch and all relational baselines are implemented with the Deep Graph Library [Wang et al., 2019b]. We train for a maximum of 200 epochs for all datasets except for Arxiv and Books, for which we train for a maximum of 1000 epochs. We use an early stopping patience of 100 and optimize with Adam.

We tune the following model-specific hyperparameters:

- **MLP**: Learning rate, weight decay, number of hidden layers, hidden layer dimension, dropout rate.
- **Label propagation**: Weighting hyperparameter, graph normalization technique, and number of propagation iterations (max 50).
- **BERT**: Learning rate, number of epochs (max 9), number of warmup steps.

[2] https://ax.dev/

Table 7.3: LSF consistently achieves state-of-the-art node classification accuracy, regardless of the graph's level of homophily.

	Heterophilous graphs				Homophilous graphs			
	Cornell	Texas	Wisc.	Actor	Wiki-CS	PubMed	Arxiv	Books
MLP	84.86	84.59	85.69	36.35	74.44	82.41	54.20	70.00
BERT	-	-	-	-	-	87.74	69.93	75.83
Label propagation	58.92	58.11	45.10	25.35	73.58	85.90	68.11	77.25
GCN	58.11	68.65	66.47	30.12	78.90	87.34	71.74	79.75
GAT	54.32	68.11	65.49	29.26	79.81	88.10	70.04	80.13
GraphSAGE	77.84	82.70	84.31	36.34	80.05	89.42	71.49	80.07
SGC	55.68	69.19	66.27	28.61	77.71	87.15	68.16	77.84
LSF	**86.22**	**86.22**	**87.25**	**36.87**	**80.73**	**90.37**	**74.57**	**81.56**

- **Standard GCNs**: Learning rate, weight decay, number of hidden layers, hidden layer dimension, dropout rate, graph normalization technique, use of self-loops, number of attention heads (GAT only), type of aggregator (GraphSAGE only), number of hops (SGC only).

7.5.2 Results and Discussion

We first compare LSF against standard GCNs, then consider more specialized architectures in the heterophily and homophily cases separately.

Comparison to standard GCNs As shown in Table 7.3, LSF achieves state-of-the-art performance on all graphs considered, providing substantial improvements over all standard GCNs and the less expressive feature-only or relation-only baselines. We observe often substantial improvements that are independent of both the graph's homophily and its size: For example, on the Arxiv graph of over one million edges, our strongest baseline (GCN) achieves 71.74% accuracy, whereas LSF achieves 74.57% accuracy, an improvement of almost 3 percentage points.

Comparison to specialized GCNs Next, we consider the more specialized baselines, which have design goals similar to LSF. In Table 7.4, we compare LSF to seven recent GCN architectures designed to handle heterophily. Remarkably, we observe that LSF outperforms all seven baselines. In Table 7.5, we compare LSF to three competitive GCN-like architectures that are designed to separate feature learning from relational learning. Again, LSF outperforms or matches these architectures. On three of the four homophilous graphs, LSF improves accuracy by one to two points over the best respective baseline. On the Books dataset, LSF comes second to DAGNN [Liu et al.,

Table 7.4: LSF consistently achieves state-of-the-art node classification accuracy on the **heterophilous** graphs, even when compared to competitive GCN architectures designed to handle heterophily. $^{\diamondsuit}$: Numbers reported by the original paper over the same splits. $^{\heartsuit}$: Numbers reported by Lim et al. [2021]. Note that the original GCNII paper did not use the Actor graph, so we reprint the number reported by Lim et al. [2021].

	Cornell	Texas	Wisc.	Actor
Geom-GCN$^{\diamondsuit}$	60.81	67.57	64.12	31.63
GCNII$^{\diamondsuit}$	76.49	77.84	81.57	34.36
H2-GCN$^{\diamondsuit}$	82.16	84.86	86.67	35.86
WRGAT$^{\diamondsuit}$	81.62	83.62	86.98	36.53
GPR-GNN$^{\heartsuit}$	68.65	76.22	75.69	33.12
MLP+GCN$^{\diamondsuit}$	84.82	83.60	86.43	36.24
LINKX$^{\diamondsuit}$	77.84	74.60	75.49	36.10
LSF	**86.22**	**86.22**	**87.25**	**36.87**

Table 7.5: LSF outperforms or matches competitive baselines designed to separate feature and relational learning on the **homophilous** graphs. $^{\diamondsuit}$: Numbers reported from the original paper over the same splits. Note that the original C&S paper used different splits for PubMed and did not use Books, so we report its performance using DGL's C&S implementation.

	Wiki-CS	PubMed	Arxiv	Books
APPNP	78.67	87.35	66.02	78.40
DAGNN	71.03	88.01	72.35	**81.67**
C&S$^{\diamondsuit}$	79.57	86.84	72.62	77.68
LSF	**80.70**	**90.37**	**74.57**	81.56

2020a] by a small difference of 0.11 points. However, DAGNN is not a competitive baseline on the other homophilous graphs, whereas LSF is consistently robust independent of the homophily level.

Error reduction Finally, to provide further evidence as to LSF's robustness across the spectrum of heterophily, we compute its error reduction relative to the best baseline per dataset in Table 7.6. We also include GraphSAGE, which is our strongest GCN competitor in terms of robustness across homophily, for comparison. On average, LSF reduces error as compared to the best baseline by 4.94%, whereas GraphSAGE increases error by 11.79%.

Table 7.6: LSF reduces error by 4.94%, on average, from the best baseline per dataset, whereas our most robust baseline GraphSAGE increases error on average by 11.79%. First three columns: Accuracy of the best baseline, GraphSAGE, and LSF per dataset. Last two columns: Error reduction of GraphSAGE and LSF, as compared to the best baseline's accuracy, per dataset.

	Accuracy			Error reduction	
	Baseline	SAGE	LSF	SAGE	LSF
Cornell	84.86	77.84	**86.22**	-46.37%	8.98%
Texas	84.86	82.70	**86.22**	-14.27%	8.98%
Wisconsin	86.98	84.31	**87.25**	-20.51%	2.07%
Actor	36.53	36.34	**36.87**	-0.30%	0.54%
Wiki-CS	80.05	80.05	**80.73**	0.00%	3.41%
PubMed	89.42	89.42	**90.37**	0.00%	8.98%
Arxiv	72.62	71.49	**74.57**	-4.13%	7.12%
Books	**81.67**	80.07	81.56	-8.73%	-0.60%

7.5.2.1 Empirical Analysis

Now that we have established the good performance of LSF, we provide further empirical characterizations of its performance. We formulate three questions:

Q1 How does LSF's ability to model homophily at a local level yield improvements in practice?

Q2 How does LSF compare to ensembles that use feature-only models *or* relational learning models, but not both?

Q3 How robust is LSF to hyperparameters?

In the remainder of this section, we answer each question in turn.

Q1: How does LSF leverage local homophily patterns? Recall from Section 7.3 that we defined the 1-hop homophily score (Equation 7.2), a node-level statistic that allows us to characterize a graph's homophily on a local level. To understand how LSF's performance is related to local homophily patterns in a graph, we plot the accuracy of LSF compared to its feature-only and relational base learners, binning nodes by 1-hop homophily score and computing accuracy of each method per bin.

Figure 7.2 shows that, as expected, the feature-only learner (MLP or BERT) consistently outperforms the relational learner (GCN) for nodes with lower 1-hop homophily, whereas the reverse is true in the higher-homophily bins. More importantly, the figures suggest that LSF interpolates between the performance of the feature-only and relational learners; this observation is consistent with the analysis of Section 7.4.2, where we showed how LSF leverages the stronger base learners to improve upon the weaker base learners.

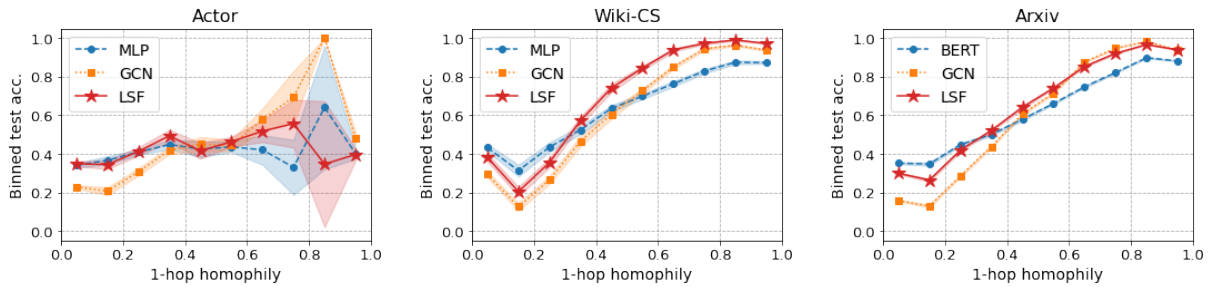


Figure 7.2: LSF discards most of the relational information in graph regions where the 1-hop homophily score is lower, and exploits the feature and relation correlations in graph regions where the 1-hop homophily score is higher. Nodes are binned by 1-hop homophily score (Equation 7.2) across 10 bins, and each method's accuracy is computed per bin at a 95 percent confidence interval.

Table 7.7: LSF flips the incorrect votes of relational ensembles under heterophily, and flips the incorrect votes of feature-only ensembles under homophily. For each direction, we provide the fraction of test nodes for which LSF flips the incorrect votes from each group of base learners, as well as the mean 1-hop homophily score (Equation 7.2) for those test nodes.

	Flip f_j^{REL} incorrect $\rightarrow$ correct		Flip f_i^{TEXT} incorrect $\rightarrow$ correct	
	% test nodes	Mean 1HH	% test nodes	Mean 1HH
Cornell	29.73%	0.326	0.0%	-
Texas	10.81%	0.000	0.0%	-
Wisconsin	23.53%	0.250	0.0%	-
Actor	15.39%	0.184	1.58%	0.295
Wiki-CS	3.03%	0.350	6.15%	0.694
PubMed	5.04%	0.392	2.14%	0.898
Arxiv	6.93%	0.339	5.70%	0.654
Books	2.81%	0.306	10.99%	0.688

Table 7.7 provides further insight into how LSF flips the incorrect majority votes made by its feature-only and relational base learners for each graph dataset we consider. As shown in Table 7.7, on the heterophilous graphs, LSF primarily corrects the votes of its relational base learners on nodes with low 1-hop homophily, whereas it corrects the votes of its feature-only base learners relatively infrequently. By contrast, on the higher-homophily graphs, LSF makes corrections to both groups of base models. The ability of LSF to correct the incorrect votes of its base models follows directly from Lemma 7.4.4, providing further empirical justification for the design of LSF.

Q2: How does LSF compare to single-modality ensembles? Since LSF is an ensembling approach, it is natural to wonder its improvements simply come from the fact that it is an ensemble—which, theoretically, could improve the performance of any model—versus the fact that it is specifically an ensemble of feature-only and relational learners. Although we justified ensembling across

Table 7.8: Ensembling a single type of base learner, either feature *or* relational, does not consistently improve accuracy across graphs and sometimes even decreases accuracy. By contrast, LSF, which ensembles feature *and* relational learners, always leads to improvement. The numbers for the best single feature and relational learners are reprinted from Table 7.3.

	Cornell	Texas	Wisc.	Actor	Wiki-CS	PubMed	Arxiv	Books
Best feature-only learner	84.86	84.59	85.69	36.35	74.44	87.74	69.93	75.83
Best feature-only ensemble	84.32	85.41	86.67	36.45	74.24	88.98	71.44	76.39
Best relational learner	77.84	82.70	84.31	36.34	80.13	89.42	70.22	80.07
Best relational learner ensemble	70.81	77.03	80.98	35.15	80.07	88.48	71.01	79.98
LSF	86.22	86.22	87.25	36.87	80.73	90.37	74.57	81.56

feature-only and relational models in Section 7.4.2, here we provide additional empirical evidence for the necessity of our design choice.

Table 7.8 compares the performances of feature-only *or* relational ensembles to LSF, using the same model selection approach that we used to construct the LSF ensemble. In most cases, we find that ensembling a single type of model either does not help accuracy or only improves it marginally. None of the feature-only or relational ensembles approaches performance comparable to LSF. In fact, in some cases ensembling a single type of model even hurts performance significantly: For example, the best single relational learning model on the heterophilous Cornell achieves 77.84% accuracy, whereas the best ensemble of relational learners that we constructed achieves 70.81% accuracy, an absolute decrease in 7.03% percentage points. By contrast, because LSF is designed with the knowledge that ensembling feature-only and relational models covers both heterophilous and homophilous structures, it consistently achieves superior performance across every graph.

7.5.2.2 Q3: How robust is LSF to hyperparameters?

Finally, to provide a brief characterization of LSF's robustness to hyperparameters, we vary each of α, N, and M in turn while keeping the other two constant. As shown in Figure 7.3, the optimal value of α is directly informed by the homophily level of the graph. For the heterophilous Actor graph, peak performance is reached with $\alpha \geq 0.7$, whereas for the homophilous Arxiv graph, peak performance is reached with $\alpha \in [0.4, 0.5]$.

Figure 7.3 also demonstrates that the number of feature-only learners does not significantly impact LSF performance, but more than three relational learners consistently hurts performance. This is likely because the predictions of the relational models are highly correlated due to their use of the graph structure, which is disadvantageous from an ensembling diversity perspective [Kuncheva and Whitaker, 2003]. We therefore recommend values of $N \in \{3, 5\}$ and $M \in \{2, 3\}$ for practical

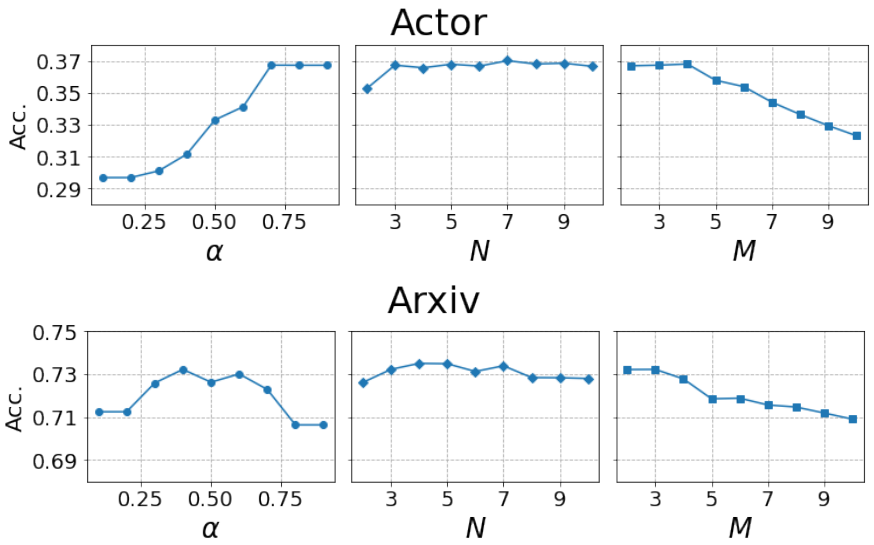


Figure 7.3: LSF's robustness to hyperparameters depends on two factors: **(1)** The weighting parameter α that controls the influence of the feature-only models, which yields better performance when inversely correlated with homophily level; and **(2)** The number of relational base models M, which should be low. For each hyperparameter being varied, we hold the other two constant.

applications of LSF.

7.6 Conclusion

In this chapter, we studied the graph learning task of node classification across the spectrum of homophily and heterophily. Motivated by the idea that an ideal approach should automatically balance the influence of structure and text depending on the homophily patterns in the graph, we presented LSF, a simple but surprisingly effective baseline for node classification that uses cross-modal ensembling. LSF models homophily on a local instead of global level for best performance in node classification by building an ensemble across graph learning and text learning models, which we demonstrate both theoretically and empirically to lead to node-level adapation depending on the graph's class distribution patterns. In experiments, we demonstrated LSF's excellent and robust performance across eight graph datasets against a host of highly competitive baselines, and provided in-depth empirical analysis to support our findings.

Part III

Conclusion

CHAPTER 8

Conclusion and Future Work

8.1 Summary

In this thesis, we proposed new methodologies and resources for text-augmented graph learning toward **(1)** knowledge representation and reasoning; and **(2)** interaction and content mining.

Knowledge representation with language models In the first part of the thesis, we focused on relational knowledge representation and reasoning with pretrained contextual language models (LMs). We set the stage for this topic in Chapter 3 by proposing a **novel taxonomy for knowledge representation in LMs** using different levels of knowledge base (KB) supervision: No KB supervision (i.e., probing the knowledge of LMs pretrained at the word level only), entity-level pretraining and fine-tuning, and relation-level pretraining and fine-tuning. For each level, we highlighted notable methodologies and findings, and made connections between NLP and graph learning, leading the way into the next two chapters.

We next introduced two novel modeling frameworks for automatic KB completion (KBC). In Chapter 4 we proposed **NegatER**, which addresses the problem of generating negative examples in commonsense knowledge bases (KBs), toward better training of discriminative commonsense KBC models. NegatER "contradicts" the commonsense knowledge patterns stored in a language model's parameters, generating new out-of-KB samples that are likely to be hard negatives. We demonstrated in our experiments that training with negative samples generated by NegatER improves the accuracy of commonsense KBC models significantly over those generated by competitive baselines. We also confirmed that NegatER-generated negative samples are deemed higher-quality than those generated by baselines in a human judgment evaluation.

We next focused on encyclopedic KBC in Chapter 5. To address the need for an encyclopedic KBC benchmark spanning structure and text, we introduced **CoDEx**, and showed its myriad advantages over existing KBC benchmarks in terms of scope and difficulty. We then proposed **CascadER**, a multi-stage ranking approach that cascades structure-only embedding models and

LMs for KBC, in order to exploit the complementary behaviors of each type of model for KBC while avoiding the inefficiencies of LMs for text ranking. We showed that CascadER achieves consistent and appreciable gains over structure-only, text-only, *and* cross-modal baselines on multiple link prediction benchmarks, including but not limited to CoDEx. We also analyzed CascadER empirically to demonstrate how it trades off effectiveness and efficiency, showing that it outperforms a highly competitive ensembling baseline while improving inference efficiency by one or more orders of magnitude.

Interaction and content mining In the second part of the thesis, we considered information retrieval and recommendation tasks that involve mining document interactions and content. In Chapter 6, we introduced the concept of the **Personal Web** as a graph of personal information objects like emails, files, and contacts. To support various downstream personal information retrieval and recommendation tasks, we proposed to efficiently learn representations of objects in the Personal Web using an integrated structure and content objective that can be updated incrementally as new data are observed. We demonstrated the strengths of Personal Webs in two recommendation tasks framed as link prediction in a graph. We first formulated a personal entity recommendation task in which we collected judgments from a small set of participants over their own data. We next devised a downstream email recipient recommendation task over a larger public dataset. In both tasks, Personal Webs outperformed diverse baselines according to various ranking metrics, suggesting that accurately modeling the unique interplay of interaction and content is key to search and recommendation over personal information collections.

Finally, in Chapter 7 we considered the task of text classification, framed as a node classification problem in a document interaction graph containing both inter-document links and document contents. We proposed Late-Stage Fusion or **LSF**, a new cross-modal ensembling approach that combines the outputs of graph- and text-based classification models. We showed in a theoretical analysis that LSF yields consistent performance regardless of the label distribution of documents in the graph. We demonstrated this theory in practice, achieving remarkably robust and excellent performance on eight diverse document interaction graph datasets, in all cases matching or outperforming state-of-the-art node classification architectures.

8.2 Vision and Future Work

There are many exciting avenues for future development in text-augmented graph learning. We conclude this thesis by outlining five concrete directions for future work.

Inductive learning With the exception of Chapter 4, in this thesis we only considered evaluation under the transductive setting, in which we assumed that the model being evaluated has seen all test entities and relationships at training time. However, an arguably more realistic evaluation setting that reflects real-world data distributional shifts is *inductive*, in which entities and relationships not observed during training are present at test time. Inductive learning is gaining interest in the graph learning community. For example, a new inductive link prediction benchmark based on our proposed CoDEx dataset (Chapter 5) has recently been introduced for inductive knowledge graph completion [Galkin et al., 2022]. Similarly, inductive node classification has been identified as an important direction for graph neural networks research, as most graph neural networks are transductive [Hamilton et al., 2017]. We expect that text-augmented graph learning in the inductive setting will rely heavily on contextual language models, as language models can encode and reason over novel sequences at test time; indeed, this is the approach we took in Chapter 4 with NegatER, as NegatER represents entities in commonsense KBs via their textual contents and encodes them with LMs. The next step is to extend similar approaches to disambiguated KBs, for example by mapping unseen entities to their closest counterparts in the training set using textual content similarity [Daza et al., 2021, Nadkarni et al., 2021].

Dense text-augmented graph representations Graph learning architectures that rely on graph propagation typically assume a set of input features for each node. When nodes correspond to documents, as in Chapters 6 and 7, such features are extracted from document contents. In this thesis we primarily considered sparse bag-of-words text features for nodes, as sparse representations are traditionally strong features for information retrieval [Luan et al., 2021] and are also efficient to process in the case of incremental model updating (Chapter 6). However, an emerging direction in information retrieval and natural language processing is to encode document contents with language models like BERT to obtain dense latent document representations, and conduct document matching and retrieval in this latent space [Karpukhin et al., 2020, Luan et al., 2021]. It has been shown that, given an appropriate training regime, such dense representations can outperform sparse representations in various text retrieval tasks [Xiong et al., 2020a]. A promising future direction in text-augmented graph learning is to develop graph-based retrieval and recommendation methods that are tailored to dense rather than sparse text representations. Such approaches will require novel training objectives that encourage representational similarity at both textual and structural levels, and will also require new techniques for efficient representation updating as the data evolve, similar to our incremental approach in Chapter 6. Toward the latter, a viable approach from transfer learning is adapters [Houlsby et al., 2019], which, given a set of learned parameters from a deep model, tune only a small fraction of these parameters as new data arrive to maintain efficiency.

Theory of cross-modal modeling We have already provided abundant empirical evidence that combining structure and text yields improvements over single-modality graph learning approaches. However, there is still ample room for further theoretical characterization of these gains. For example, we provided a theoretical justification for the superior performance of our cross-modal ensembling approach LSF over single-modality ensembles for node classification in Chapter 7. A natural next step is to develop similar theoretical foundations for the improvements achieved by CascadER in Chapter 5, as we observed similarly large gains over single-modality approaches with CascadER in the context of link prediction. We believe that future work in combining structure and content will require establishing the conditions under which cross-modal modeling can provably boost accuracy over single-modality graph learning approaches, in order to unify theory and practice.

Multi-objective learning In this thesis, we focused primarily on improving accuracy in graph learning tasks. However, given the prevalence of the problems we consider, we envision future work in text-augmented graph learning incorporating complex objectives beyond accuracy, toward robust performance in practical real-world settings. In tasks involving large-scale data, an important future direction is to incorporate explicit resource cost penalties or latency constraints, for example within the multi-stage CascadER framework proposed in Chapter 5, in order to better control the tradeoff between effectiveness and efficiency. In tasks involving user interactions, for example the personal search and recommendation settings considered in Chapter 6, future directions include defining and optimizing objectives for user satisfaction, privacy, and exposure to diverse content, while still developing novel and effective interaction and content models.

Fairness and ethics As demonstrated throughout Part I, pretrained contextual language models have contributed to enormous progress in knowledge representation and reasoning. However, at their core, they are conditional probability estimators of words in a corpus, not necessarily factually faithful or grounded knowledge representations [Bender and Koller, 2020, Bisk et al., 2020]. Inasmuch as they are able to express factual content in text, they can also express harmful content like offensive language and social stereotypes [Bender et al., 2021]. While there is intense interest within NLP in defining, understanding, and improving fairness and ethics in LMs [Blodgett et al., 2020], such work is relatively separate from the goals and methodologies considered within this thesis, even though goals like commonsense acquisition (Chapter 4) are fundamentally tied to cultural biases, social stereotypes, and human values. Given that LMs are now a major component of language processing systems, an important future direction is to understand and mitigate potential biases in LMs as they apply to text-augmented graph learning, toward more socially-aware and human-aligned machine learning systems.

APPENDIX A

NegatER annotation instructions

In this section we provide the annotation instructions for Ch. 4.7.

A.1 Task definition

In this task you will judge a set of statements based on how grammatical, truthful, and consistent they are. Each statement is given in [head phrase, relation, tail phrase] form. The criteria are as follows:

- **Grammar**: Our definition of grammar refers to whether each statement follows the grammar rules we provide for its relation type. We do not include proper use of punctuation (e.g., commas, apostrophes) or articles (e.g., "the", "a", "this") in our definition of grammar. The choices are "correct", "partially correct or unsure", and "incorrect". (*Note: in our analyses we binarize these choices, considering "partially correct" and "incorrect" as the same.*)
- **Truthfulness**: Our definition of truthfulness refers to how often you believe the whole statement holds true. The choices are: "always true", "mostly true", "sometimes true", and "never true".
- **Consistency**: We define "consistency" as the degree to which the head and tail phrases are consistent in terms of the topic, theme, or goal that they refer to. For example, the phrases "football" and "baseball" are highly consistent because they both refer to team sports, whereas the phrases "football" and "cactus" are not consistent. The choices are: "highly consistent", "somewhat consistent", "a little consistent", and "not consistent at all".

You may fill your answers in any order. For example, you might find it helpful to judge the grammar of all statements first, then the truthfulness, then the consistency. Some of the statements are subjective and there is not always a "right" answer, especially for the consistency criterion. If you are unsure of a word or reference, you may use Google or other search engines. You may also explain your reasoning/interpretation in the optional Notes box.

A.2 Examples and explanations

HasPrerequisite The `HasPrerequisite` relation describes prerequisites or pre-conditions for actions or states of being. It requires a verb phrase (an action or a state of being) in the head slot and a verb phrase or noun phrase in the tail slots. Examples:

- ("pay bill", `HasPrerequisite`, "have money")
  - Grammar: correct
  - Truthfulness: always true
  - Consistency: highly consistent
- ("purchase a cellular phone", `HasPrerequisite`, "study")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: not consistent at all
- ("paint your house", `HasPrerequisite`, "purple")
  - Grammar: incorrect
  - Truthfulness: never true
  - Consistency: a little consistent (*Our interpretation: Painting your house involves choosing a color, so the statement could be construed as a little consistent, even though it's grammatically incorrect.*)
- ("eat", `HasPrerequisite`, "send them to their room")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: not consistent at all

HasProperty The `HasProperty` relation describes properties of actions or objects. It requires a verb phrase or noun phrase in the head slot and a description in the tail slot. Examples:

- ("school bus", `HasProperty`, "yellow")
  - Grammar: correct

  - Truthfulness: mostly true (*Our interpretation: Yellow school buses are very common in the USA and Canada, but not all school buses are yellow.*)
  - Consistency: highly consistent

- ("basketball", `HasProperty`, "round")
  - Grammar: correct
  - Truthfulness: always true
  - Consistency: highly consistent

- ("pilot", HasProperty, "land airplane")
  - Grammar: incorrect
  - Truthfulness: never true
  - Consistency: highly consistent (*Our interpretation: While pilots do land airplanes, the HasProperty relation requires a description in the tail slot, so it's not grammatically correct or truthful.*)

- ("gross domestic product", `HasProperty`, "abbreviated to CTBT")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: a little consistent (*Our interpretation: The gross domestic product does have a well-known abbreviation ("GDP"), so this statement could be construed as a little consistent.*)

HasSubevent The `HasSubevent` relation describes sub-events or components of larger events. It requires an event (verb phrase or noun phrase) in the head slot and an event in the tail slot. Examples:

- ("lying", `HasSubevent`, "you feel guilty")
  - Grammar: correct
  - Truthfulness: mostly true (*Our interpretation: Lying often causes guilt in people, although the amount of guilt depends on the person.*)
  - Consistency: highly consistent

- ("relax", `HasSubevent`, "vegetable")

  - Grammar: incorrect
  - Truthfulness: never true
  - Consistency: not consistent at all

- ("drink coffee", `HasSubevent`, "water may get into your nose")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: a little consistent (*Our interpretation: Drinking coffee doesn't cause water to get into your nose, but coffee and water are both drinkable liquids, so we think this statement is a little consistent.*)

ReceivesAction The `ReceivesAction` relation describes actions that apply to objects or other actions. It requires a verb phrase or noun phrase in the head slot and an action in the tail slot. Examples:

- ("book", `ReceivesAction`, "write by person")
  - Grammar: correct
  - Truthfulness: always true
  - Consistency: highly consistent

- ("most watches", `ReceivesAction`, "rhyme with piano")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: not consistent at all

- ("oil", `ReceivesAction`, "grow in field")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: a little consistent (*Our interpretation: Since oil is a natural resource similar to other things that are grown in fields, we could see this statement being a little consistent (it's a stretch though).*)

- ("violin", `ReceivesAction`, "play with a puck")

  - Grammar: correct
  - Truthfulness: never true
  - Consistency: somewhat consistent (*Our interpretation: Violins are indeed played, but with a bow, not a puck.*)

UsedFor The `UsedFor` relation describes the uses of objects or actions. It requires a verb phrase or noun phrase in the head and tail slots. Examples:

- ("shoes", `UsedFor`, "protecting feet")
  - Grammar: correct
  - Truthfulness: always true
  - Consistency: highly consistent
- ("tying your shoelace", `UsedFor`, "smart")
  - Grammar: incorrect
  - Truthfulness: never true
  - Consistency: not consistent at all
- ("swimming", `UsedFor`, "traveling on land")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: somewhat consistent (*Our interpretation: This statement is somewhat consistent because swimming and traveling on land are both means of movement.*)
- ("bush", `UsedFor`, "wrestling on")
  - Grammar: correct
  - Truthfulness: never true
  - Consistency: not consistent at all

BIBLIOGRAPHY

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Large scale knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, 2021. URL https://arxiv.org/pdf/2010.12688.pdf.

Kian Ahrabian, Aarash Feizi, Yasmin Salehi, William L. Hamilton, and Avishek Joey Bose. Structure aware negative sampling in knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6093–6101, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.492. URL https://www.aclweb.org/anthology/2020.emnlp-main.492.

Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1995–2010, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3318464.3380599.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: a python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research*, 22(82):1–6, 2021. URL https://jmlr.org/papers/v22/20-825.html.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, New Orleans - Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-3011. URL https://www.aclweb.org/anthology/N18-3011.

Amith Ananthram, Emily Allaway, and Kathleen McKeown. Event-guided denoising for multilingual relation learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1505–1512, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.131. URL https://www.aclweb.org/anthology/2020.coling-main.131.

Hiba Arnaout, Simon Razniewski, and Gerhard Weikum. Enriching knowledge bases with interesting negative statements. In *Automated Knowledge Base Construction*, 2020. URL https://openreview.net/pdf?id=pSLmyZKaS.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1522. URL https://aclanthology.org/D19-1522.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. Multi-relational poincaré graph embeddings. In *Advances in Neural Information Processing Systems*, 2019b. URL https://proceedings.neurips.cc/paper/2019/file/f8b932c70d0b2e6bf071729a4fa68dfc-Paper.pdf.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1279. URL https://www.aclweb.org/anthology/P19-1279.

Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P08-1004.

Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. A2N: Attending to neighbors for knowledge graph inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4387–4392, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1431. URL https://aclanthology.org/P19-1431.

Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E Schapire. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998. URL https://www.jstor.org/stable/120016.

Vladimir Batagelj and Matjaž Zaveršnik. Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5(2): 129–145, 2011. URL https://link.springer.com/content/pdf/10.1007/s11634-010-0079-y.pdf.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. URL https://arxiv.org/pdf/1806.01261.

Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association*

for Computational Linguistics, pages 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL https://www.aclweb.org/anthology/2020.acl-main.463.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Michael Bendersky, Xuanhui Wang, Donald Metzler, and Marc Najork. Learning from user interactions in personal search via attribute parameterization. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 791–799, 2017. URL https://dl.acm.org/doi/pdf/10.1145/3018661.3018712.

Jan R Benetka, John Krumm, and Paul N Bennett. Understanding context for tasks and activities. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 133–142, 2019. URL https://dl.acm.org/doi/pdf/10.1145/3295750.3298929.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003. URL https://www.jmlr.org/papers/volume3/tmp/bengio03a.pdf.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.703. URL https://www.aclweb.org/anthology/2020.emnlp-main.703.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. URL https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.485. URL https://aclanthology.org/2020.acl-main.485.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008. URL https://dl.acm.org/doi/pdf/10.1145/1376616.1376746.

Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Andreas Bender, Charles Tapley Hoyt, and William Hamilton. A review of biomedical datasets relating to drug

discovery: A knowledge graph perspective. *arXiv preprint arXiv:2102.10062*, 2021. URL https://arxiv.org/pdf/2102.10062.pdf.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems*, pages 2787–2795, 2013. URL https://dl.acm.org/doi/10.5555/2999792.2999923.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1470. URL https://www.aclweb.org/anthology/P19-1470.

Jan A. Botha, Zifei Shan, and Daniel Gillick. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.630. URL https://www.aclweb.org/anthology/2020.emnlp-main.630.

Guillaume Bouchard, Sameer Singh, and Theo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Symposium Series*, 2015. URL https://www.aaai.org/ocs/index.php/SSS/SSS15/paper/viewPaper/10257.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/download/6242/6098.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. URL https://link.springer.com/content/pdf/10.1007/BF00058655.pdf.

Samuel Broscheit. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1063. URL https://www.aclweb.org/anthology/K19-1063.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. Can we predict new facts with open knowledge graph embeddings? a benchmark for open link prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2308, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.209. URL https://www.aclweb.org/anthology/2020.acl-main.209.

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. LibKGE - a knowledge graph embedding library for reproducible research. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 165–174, Online, October 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.22. URL https://aclanthology.org/2020.emnlp-demos.22.

Adam S Brown and Chirag J Patel. A standard database for drug repositioning. *Scientific data*, 4 (1):1–7, 2017. URL https://www.nature.com/articles/sdata201729.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Ivan Brugere, Brian Gallagher, and Tanya Y Berger-Wolf. Network structure inference, a survey: Motivations, methods, and applications. *ACM Computing Surveys (CSUR)*, 51(2):1–39, 2018. URL https://dl.acm.org/doi/pdf/10.1145/3154524.

Vannevar Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, 1945. URL http://s3data.computerhistory.org/chess/as-we-may-think.bush-vannevar.1945.atlantic-monthly.062303004.pdf.

Liwei Cai and William Yang Wang. KBGAN: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1470–1480, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1133. URL https://www.aclweb.org/anthology/N18-1133.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.146. URL https://aclanthology.org/2021.acl-long.146.

Haotian Chen, Xi Li, Andrej Zukov Gregoric, and Sahil Wadhwa. Contextualized end-to-end neural entity linking. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 637–642, Suzhou, China, December 2020a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.aacl-main.64.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR, 2020b. URL http://proceedings.mlr.press/v119/chen20v/chen20v.pdf.

Ruey-Cheng Chen, Luke Gallagher, Roi Blanco, and J Shane Culpepper. Efficient cost-aware cascade ranking in multi-stage retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 445–454, 2017. URL https://dl.acm.org/doi/pdf/10.1145/3077136.3080819.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf?id=n6jl7fLxrP.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3882–3890, 2020. URL https://www.ijcai.org/Proceedings/2020/0537.pdf.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.207. URL https://aclanthology.org/2020.acl-main.207.

Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. Combining pre-trained language models and structured knowledge. *arXiv preprint arXiv:2101.12294*, 2021. URL https://arxiv.org/pdf/2101.12294.pdf.

Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. Random walks in recommender systems: exact computation and simulations. In *Proceedings of the 23rd international conference on world wide web*, pages 811–816, 2014. URL https://dl.acm.org/doi/pdf/10.1145/2567948.2579244.

Sergio Currarini, Matthew O Jackson, and Paolo Pin. An economic model of friendship: Homophily, minorities, and segregation. *Econometrica*, 77(4):1003–1045, 2009. URL https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA7528.

Jeff Da and Jungo Kasai. Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 1–12, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6001. URL https://www.aclweb.org/anthology/D19-6001.

Jeff Da, Ronan Le Bras, Ximing Lu, Yejin Choi, and Antoine Bosselut. Analyzing commonsense emergence in few-shot knowledge models. In *3rd Conference on Automated Knowledge Base Construction*, 2021. URL https://openreview.net/pdf?id=StHCELh9PVE.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/pdf?id=Syg-YfWCW.

Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015. URL https://dl.acm.org/doi/pdf/10.1145/2701413.

Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *AI magazine*, 14(1):17–17, 1993. URL https://ojs.aaai.org/index.php/aimagazine/article/view/1029.

Joe Davison, Joshua Feldman, and Alexander Rush. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1109. URL https://www.aclweb.org/anthology/D19-1109.

Daniel Daza, Michael Cochez, and Paul Groth. Inductive entity representations from text via link prediction. In *Proceedings of the Web Conference 2021*, pages 798–808, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3442381.3450141.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf?id=5k8F6UU39V.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990. URL https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11573.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144. ACM, 2017.

Mark Dredze and Hanna Wallach. User models for email activity management. In *IUI Workshop on Ubiquitous User Modeling*, 2008. URL https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.1223&rep=rep1&type=pdf.

Mark Dredze, Tessa Lau, and Nicholas Kushmerick. Automatically classifying emails into activities. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 70–77, 2006. URL https://dl.acm.org/doi/pdf/10.1145/1111449.1111471.

Philipp Dufter, Nora Kassner, and Hinrich Schütze. Static embeddings as efficient knowledge bases? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, Online, June 2021. Association for Computational Linguistics. URL https://arxiv.org/pdf/2104.07094.pdf.

Susan Dumais, Edward Cutrell, Jonathan J Cadiz, Gavin Jancke, Raman Sarin, and Daniel C Robbins. Stuff i've seen: a system for personal information retrieval and re-use. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72–79, 2003. URL https://dl.acm.org/doi/10.1145/2888422.2888425.

Susan Dumais, Edward Cutrell, Raman Sarin, and Eric Horvitz. Implicit queries (iq) for contextualized search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 594–594, 2004. URL https://dl.acm.org/doi/pdf/10.1145/1008992.1009137.

Takuma Ebisu and Ryutaro Ichise. Toruse: Knowledge graph embedding on a lie group. In *Thirty-second AAAI conference on artificial intelligence*, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewFile/16227/15885.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410. URL https://aclanthology.org/2021.tacl-1.60.

Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020. doi: 10.1162/tacl_a_00298. URL https://www.aclweb.org/anthology/2020.tacl-1.3.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1142.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.400. URL https://www.aclweb.org/anthology/2020.emnlp-main.400.

Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek. Predicting completeness in knowledge bases. In *Proceedings of the Tenth ACM International Conference on*

Web Search and Data Mining, pages 375–383, 2017. URL https://dl.acm.org/doi/pdf/10.1145/3018661.3018739.

Mikhail Galkin, Max Berrendorf, and Charles Tapley Hoyt. An open challenge for inductive link prediction on knowledge graphs. *arXiv preprint arXiv:2203.01520*, 2022. URL https://arxiv.org/pdf/2203.01520.

Luke Gallagher, Ruey-Cheng Chen, Roi Blanco, and J Shane Culpepper. Joint optimization of cascade ranking models. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 15–23, 2019. URL https://dl.acm.org/doi/pdf/10.1145/3289600.3290986.

Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. Composing relationships with translations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 286–290, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1034. URL https://aclanthology.org/D15-1034.

Martin Gartmeier, Johannes Bauer, Hans Gruber, and Helmut Heid. Negative knowledge: Understanding professional learning and expertise. *Vocations and Learning*, 1(2):87–103, 2008. URL https://link.springer.com/article/10.1007/s12186-008-9006-1.

Victor M González and Gloria Mark. c̈onstant, constant, multi-tasking craziness̈: Managing multiple working spheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, 2004. URL https://dl.acm.org/doi/pdf/10.1145/985692.985707.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. URL https://www.deeplearningbook.org/.

Leo A Goodman. Snowball sampling. *The annals of mathematical statistics*, pages 148–170, 1961. URL https://www.jstor.org/stable/2237615.

Catherine Grevet, David Choi, Debra Kumar, and Eric Gilbert. Overload is overloaded: email in the age of gmail. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 793–802, 2014. URL https://dl.acm.org/doi/pdf/10.1145/2556288.2557013.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016. URL https://dl.acm.org/doi/pdf/10.1145/2939672.2939754.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3458754.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108, 2020. doi: 10.1162/tacl_a_00302. URL https://doi.org/10.1162/tacl_a_00302.

Lingbing Guo, Zequn Sun, and Wei Hu. Learning to exploit long-term relational dependencies in knowledge graphs. In *International Conference on Machine Learning*, pages 2505–2514. PMLR, 2019. URL http://proceedings.mlr.press/v97/guo19c/guo19c.pdf.

Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. Semantically smooth knowledge graph embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 84–94, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1009. URL https://aclanthology.org/P15-1009.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11918/11777.

Kelvin Guu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1038. URL https://aclanthology.org/D15-1038.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR, 2020. URL http://proceedings.mlr.press/v119/guu20a/guu20a.pdf.

Alon Y Halevy, Oren Etzioni, AnHai Doan, Zachary G Ives, Jayant Madhavan, Luke K McDowell, and Igor Tatarinov. Crossing the structure chasm. In *Conference on Innovative Data Systems Research*, 2003. URL http://cidrdb.org/cidr2003/program/p11.pdf.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011. URL https://epubs.siam.org/doi/pdf/10.1137/090771806.

William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020. URL https://www.morganclaypool.com/doi/pdfplus/10.2200/S01045ED1V01Y202009AIM046.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017. URL https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

Adi Haviv, Jonathan Berant, and Amir Globerson. BERTese: Learning to speak to BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3618–3623, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.316.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.207. URL https://www.aclweb.org/anthology/2020.findings-emnlp.207.

Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.153.

Geoffrey E Hinton. Learning distributed representations of concepts. In *CogSci*, 1986. URL http://www.cs.toronto.edu/~hinton/absps/families.pdf.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a/houlsby19a.pdf.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020. URL https://proceedings.neurips.cc/paper/2020/file/fb60d411a5c5b72b2e7d3527cfc84fd0-Paper.pdf.

Luyang Huang, Lingfei Wu, and Lu Wang. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.457. URL https://aclanthology.org/2020.acl-main.457.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. Combining label propagation and simple models out-performs graph neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf?id=8E1-f3VhX1o.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/pdf/1905.01969.pdf.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16792.

Filip Ilievski, Pedro Szekely, and Bin Zhang. Cskg: The commonsense knowledge graph. *European Semantic Web Conference*, 2021. URL https://link.springer.com/chapter/10.1007/978-3-030-77385-4_41.

Ruben Interian and Celso C Ribeiro. An empirical investigation of network polarization. *Applied Mathematics and Computation*, 339:651–662, 2018. URL https://www.sciencedirect.com/science/article/pii/S0096300318306325.

Stanislaw Jastrzebski, Dzmitry Bahdanau, Seyedarian Hosseini, Michael Noukhovitch, Yoshua Bengio, and Jackie Cheung. Commonsense mining as knowledge base completion? a study on the impact of novelty. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 8–16, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-1002. URL https://www.aclweb.org/anthology/W18-1002.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1067. URL https://aclanthology.org/P15-1067.

Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1115.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020. URL https://arxiv.org/pdf/2002.00388.pdf.

Yantao Jia, Yuanzhuo Wang, Hailun Lin, Xiaolong Jin, and Xueqi Cheng. Locally adaptive translation for knowledge graph embedding. In *Thirtieth AAAI conference on artificial intelligence*, 2016. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/viewPDFInterstitial/12018/11694.

Liwei Jiang, Antoine Bosselut, Chandra Bhagavatula, and Yejin Choi. "I'm not mad": Commonsense implications of negation and contradiction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4380–4397, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.346. URL https://aclanthology.org/2021.naacl-main.346.

Xiaotian Jiang, Quan Wang, and Bin Wang. Adaptive convolution for multi-relational learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 978–987, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1103. URL https://aclanthology.org/N19-1103.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.479. URL https://www.aclweb.org/anthology/2020.emnlp-main.479.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020b. doi: 10.1162/tacl_a_00324. URL https://aclanthology.org/2020.tacl-1.28.

Di Jin, Mark Heimann, Ryan A. Rossi, and Danai Koutra. Node2bits: Compact time- and attribute-aware node representations for user stitching. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019a. URL https://link.springer.com/chapter/10.1007/978-3-030-46150-8_29.

Di Jin, Mark Heimann, Tara Safavi, Mengdi Wang, Wei Lee, Lindsay Snider, and Danai Koutra. Smart roles: Inferring professional roles in email networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2923–2933, 2019b. URL https://dl.acm.org/doi/pdf/10.1145/3292500.3330735.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. URL https://ieeexplore.ieee.org/iel7/6687317/7153538/08733051.pdf.

William Jones, Jesse David Dinneen, Robert Capra, Anne R. Diekema, and Manuel A. Pérez-Quiñones. Personal information management. 2017. doi: 10.1081/E-ELIS4-120053695. URL https://www.routledgehandbooks.com/doi/10.1081/E-ELIS4-120053695.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://www.aclweb.org/anthology/2020.tacl-1.5.

Lucie-Aimée Kaffee, Alessandro Piscopo, Pavlos Vougiouklis, Elena Simperl, Leslie Carr, and Lydia Pintscher. A glimpse into babel: an analysis of multilinguality in wikidata. In *Proceedings of the 13th International Symposium on Open Collaboration*, pages 1–5, 2017. URL https://dl.acm.org/doi/pdf/10.1145/3125433.3125465.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.698. URL https://www.aclweb.org/anthology/2020.acl-main.698.

Nora Kassner, Benno Krojer, and Hinrich Schütze. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.45. URL https://www.aclweb.org/anthology/2020.conll-1.45.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. Multilingual lama: Investigating knowledge in multilingual pretrained language models. In *The 16th Conference of the European Chapter of the Association for Computational Linguistics*, 2021. URL https://arxiv.org/pdf/2102.00894.

Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/file/b2ab001909a8a6f04b51920306046ce5-Paper.pdf.

Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference, AAAI-06/IAAI-06*, pages 381–388, 2006. URL https://www.aaai.org/Papers/AAAI/2006/AAAI06-061.pdf.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.171. URL https://www.aclweb.org/anthology/2020.findings-emnlp.171.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3397271.3401075.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.153. URL https://aclanthology.org/2020.coling-main.153.

Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/pdf?id=SJU4ayYgl.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/pdf?id=H1gL-2A9Ym.

Vid Kocijan and Thomas Lukasiewicz. Knowledge base completion meets transfer learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6521–6533, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.524. URL https://aclanthology.org/2021.emnlp-main.524.

Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001. URL https://www.aeaweb.org/articles?id=10.1257/jep.15.4.143.

Abdullatif Köksal and Arzucan Özgür. The RELX dataset and matching the multilingual blanks for cross-lingual relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 340–350, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.32. URL https://www.aclweb.org/anthology/2020.findings-emnlp.32.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1050. URL https://www.aclweb.org/anthology/K18-1050.

Daphne Koller, Nir Friedman, Sašo Džeroski, Charles Sutton, Andrew McCallum, Avi Pfeffer, Pieter Abbeel, Ming-Fai Wong, Chris Meek, Jennifer Neville, et al. *Introduction to statistical relational learning*. MIT press, 2007.

Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3075–3084. ACM, 2014. URL https://dl.acm.org/doi/pdf/10.1145/2556288.2557238.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):

181–207, 2003. URL https://link.springer.com/content/pdf/10.1023/A:1022859003006.pdf.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2863–2872. PMLR, 2018. URL http://proceedings.mlr.press/v80/lacroix18a/lacroix18a.pdf.

Jonathan Lajus, Luis Galárraga, and Fabian Suchanek. Fast and exact rule mining with amie 3. In *European Semantic Web Conference*, pages 36–52. Springer, 2020. URL https://link.springer.com/chapter/10.1007/978-3-030-49461-2_3.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. URL https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/building-machines-that-learn-and-think-like-people/A9535B1D745A0377E16C590E14B94993.

Jonathan Larson, Darren Edge, Nathan Evans, and Christopher White. Making sense of search: Using graph embedding and visualization to transform query understanding. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3334480.3375233.

Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.deelio-1.5. URL https://www.aclweb.org/anthology/2020.deelio-1.5.

Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995. URL https://dl.acm.org/doi/pdf/10.1145/219717.219745.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.86.

Furong Li, Xin Luna Dong, Anno Langen, and Yang Li. Knowledge verification for long-tail verticals. *Proceedings of the VLDB Endowment*, 10(11):1370–1381, 2017. URL https://dl.acm.org/doi/pdf/10.14778/3137628.3137646.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1137. URL https://www.aclweb.org/anthology/P16-1137.

David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007. URL https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.20591.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34, 2021. URL https://proceedings.neurips.cc/paper/2021/file/ae816a80e4c1c56caa2eb4e1819cbb2f-Paper.pdf.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6862–6868, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.557. URL https://www.aclweb.org/anthology/2020.emnlp-main.557.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021. URL https://www.morganclaypool.com/doi/pdfplus/10.2200/S01123ED1V01Y202108HLT053.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1362. URL https://aclanthology.org/D18-1362.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 705–714, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. doi: 10.18653/v1/D15-1082. URL https://aclanthology.org/D15-1082.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial*

intelligence, volume 15, pages 2181–2187, 2015b. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewFile/9571/9523.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning with entities, attributes and relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2866–2872, 2016. URL https://dl.acm.org/doi/abs/10.5555/3060832.3061022.

Jeffrey Ling, Nicholas FitzGerald, Zifei Shan, Livio Baldini Soares, Thibault Févry, David Weiss, and Tom Kwiatkowski. Learning cross-context entity representations from text. *arXiv preprint arXiv:2001.03765*, 2020. URL https://arxiv.org/pdf/2001.03765.pdf.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. Analogical inference for multi-relational embeddings. In *International conference on machine learning*, pages 2168–2178. PMLR, 2017. URL http://proceedings.mlr.press/v70/liu17d/liu17d.pdf.

Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004. URL https://link.springer.com/content/pdf/10.1023/B:BTTJ.0000047600.45421.6d.pdf.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 338–348, 2020a. URL https://dl.acm.org/doi/pdf/10.1145/3394486.3403076.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021a. URL https://arxiv.org/pdf/2107.13586.pdf.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908, 2020b. URL https://ojs.aaai.org/index.php/AAAI/article/download/5681/5537.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021b. URL https://arxiv.org/pdf/2103.10385.pdf.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021c. URL https://www.aaai.org/AAAI21Papers/AAAI-4301.LiuY.pdf.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/pdf/1907.11692.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1335. URL https://www.aclweb.org/anthology/P19-1335.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021. doi: 10.1162/tacl_a_00369. URL https://aclanthology.org/2021.tacl-1.20.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021a. URL https://www.aaai.org/AAAI21Papers/AAAI-1059.MaK.pdf.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021b. URL https://arxiv.org/pdf/2106.06134.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *Conference on Innovative Data Systems Research*, 2014. URL http://cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. Commonsense knowledge base completion with structural and semantic context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2925–2933, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/download/5684/5540.

Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1577–1580, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3397271.3401266.

Julian Mcauley and Jure Leskovec. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):1–28, 2014. URL https://dl.acm.org/doi/pdf/10.1145/2556612.

Alexa T McCray. An upper-level ontology for the biomedical domain. *Comparative and functional genomics*, 4(1):80–84, 2003. URL https://onlinelibrary.wiley.com/doi/pdf/10.1002/cfg.255.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001. URL https://www.annualreviews.org/doi/full/10.1146/annurev.soc.27.1.415.

Edgar Meij, Tara Safavi, Chenyan Xiong, Gianluca Demartini, Miriam Redi, and Fatma Özcan. Proceedings of the kg-bias workshop 2020 at akbc 2020. *arXiv preprint arXiv:2007.11659*, 2020. URL https://arxiv.org/abs/2007.11659.

Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020. URL https://arxiv.org/pdf/2007.02901.pdf.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013. URL https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

Marvin Minsky. A framework for representing knowledge, 1974.

Marvin Minsky. Negative expertise. *International Journal of Expert Systems*, 7(1):13–19, 1994. URL https://web.media.mit.edu/~minsky/papers/NegExp.mss.txt.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P09-1113.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018. URL https://dl.acm.org/doi/pdf/10.1145/3191513.

Rahul Nadkarni, David Wadden, Iz Beltagy, Noah Smith, Hannaneh Hajishirzi, and Tom Hope. Scientific language models for biomedical knowledge base completion: An empirical study. In *3rd Conference on Automated Knowledge Base Construction*, 2021. URL https://openreview.net/pdf?id=4Exq_UvWKY8.

Galileo Namata, Ben London, Lise Getoor, Bert Huang, and UMD EDU. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8, page 1, 2012. URL https://people.cs.vt.edu/~bhuang/papers/namata-mlg12.pdf.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1466. URL https://aclanthology.org/P19-1466.

Jennifer Neville and David Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8(3), 2007. URL https://www.jmlr.org/papers/volume8/neville07a/neville07a.pdf.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2053. URL https://aclanthology.org/N18-2053.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2180–2189, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1226. URL https://aclanthology.org/N19-1226.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. STransE: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 460–466, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1054. URL https://aclanthology.org/N16-1054.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-2074. URL https://www.aclweb.org/anthology/P16-2074.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, 2011. URL https://icml.cc/2011/papers/438_icmlpaper.pdf.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015. URL https://ieeexplore.ieee.org/document/7358050.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. URL https://ojs.aaai.org/index.php/AAAI/article/view/10314/10173.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*, 2019. URL https://arxiv.org/pdf/1910.14424.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=S1ldO2EFPr.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=S1e2agrFvS.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34, 2021. URL https://proceedings.neurips.cc/paper/2021/file/5c04925674920eb58467fb52ce4ef728-Paper.pdf.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014. URL https://dl.acm.org/doi/pdf/10.1145/2623330.2623732.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1005. URL https://www.aclweb.org/anthology/D19-1005.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://www.aclweb.org/anthology/D19-1250.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*, 2020. URL https://openreview.net/forum?id=025X0zPfn.

Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. Revisiting evaluation of knowledge base completion models. In *Automated Knowledge Base Construction*, 2020. URL https://openreview.net/pdf?id=1uufzxsxfL.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.71. URL https://www.aclweb.org/anthology/2020.findings-emnlp.71.

Ashequl Qadir, Michael Gamon, Patrick Pantel, and Ahmed Hassan Awadallah. Activity modeling in email. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1452–1462, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1171. URL https://aclanthology.org/N16-1171.

Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.410. URL https://aclanthology.org/2021.naacl-main.410.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3350–3363, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.260. URL https://aclanthology.org/2021.acl-long.260.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26, 2020. URL https://link.springer.com/content/pdf/10.1007/s11431-020-1647-3.pdf.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL http://www.persagen.com/files/misc/radford2019language.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020. URL https://www.jmlr.org/papers/volume21/20-074/20-074.pdf.

Simon Razniewski and Priyanka Das. Structured knowledge: Have we made progress? an extrinsic study of kb coverage over 19 years. In *Proceedings of the 29th ACM International Conference*

on Information & Knowledge Management, volume 54, pages 3317–3320, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3340531.3417447.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.nlp4convai-1.20. URL https://aclanthology.org/2021.nlp4convai-1.20.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://www.aclweb.org/anthology/2020.emnlp-main.437.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8: 842–866, 2021. URL https://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00349.

Julien Romero, Simon Razniewski, Koninika Pal, Jeff Z. Pan, Archit Sakhadeo, and Gerhard Weikum. Commonsense properties from query logs and question answering forums. In *CIKM*, pages 1411–1420, 2019. URL https://dl.acm.org/doi/pdf/10.1145/3357384.3357955.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*, 2020. URL https://arxiv.org/pdf/2007.00655.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=BkxSmlBFvr.

Tara Safavi and Danai Koutra. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8328–8350, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.669. URL https://aclanthology.org/2020.emnlp-main.669.

Tara Safavi and Danai Koutra. Relational World Knowledge Representation in Contextual Language Models: A Review. In *Proceedings of the 2021 Conference on Empirical Methods*

in Natural Language Processing, pages 1053–1067, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.81. URL https://aclanthology.org/2021.emnlp-main.81.

Tara Safavi, Adam Fourney, Robert Sim, Marcin Juraszek, Shane Williams, Ned Friend, Danai Koutra, and Paul N Bennett. Toward activity discovery in the personal web. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 492–500, 2020. URL https://gemslab.github.io/papers/safavi-2020-toward.pdf.

Tara Safavi, Jing Zhu, and Danai Koutra. NegatER: Unsupervised Discovery of Negatives in Commonsense Knowledge Bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5633–5646, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-main.456.

Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. Commonsense knowledge base completion and generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 141–150, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1014. URL https://www.aclweb.org/anthology/K18-1014.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035, 2019. URL https://ojs.aaai.org/index.php/AAAI/article/view/4160/4038.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1036. URL https://aclanthology.org/D15-1036.

Jianqiang Shen, Lida Li, Thomas G Dietterich, and Jonathan L Herlocker. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 86–92. ACM, 2006. URL https://dl.acm.org/doi/pdf/10.1145/1111449.1111473.

Tao Shen, Yi Mao, Pengcheng He, Guodong Long, Adam Trischler, and Weizhu Chen. Exploiting structured knowledge in text via graph-guided representation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8980–8994, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.722. URL https://www.aclweb.org/anthology/2020.emnlp-main.722.

Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950. URL https://www.jstor.org/stable/pdf/2236561.pdf.

Baoxu Shi and Tim Weninger. Proje: Embedding projection for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. URL https://ojs.aaai.org/index.php/AAAI/article/view/10677/10536.

Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2016. URL https://ieeexplore.ieee.org/iel7/69/7775118/07536145.pdf.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.346. URL https://www.aclweb.org/anthology/2020.emnlp-main.346.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.373. URL https://www.aclweb.org/anthology/2020.emnlp-main.373.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013. URL https://proceedings.neurips.cc/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf.

Robyn Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686, 2012. URL http://lrec-conf.org/proceedings/lrec2012/pdf/1072_Paper.pdf.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. URL https://ojs.aaai.org/index.php/AAAI/article/download/11164/11023.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007. URL https://dl.acm.org/doi/pdf/10.1145/1242572.1242667.

Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3 (2):1–159, 2012. URL https://www.morganclaypool.com/doi/abs/10.2200/s00433ed1v01y201207dmk005.

Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*,

4(11):992–1003, 2011. URL https://dl.acm.org/doi/pdf/10.14778/3402707.3402736.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/download/6428/6284.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/pdf/ebe1e67180cf2643b41b4888108081fc1538ce11.pdf.

Susheel Suresh, Vinith Budde, Jennifer Neville, Pan Li, and Jianzhu Ma. Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3447548.3467373.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://www.aclweb.org/anthology/N19-1421.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758, 2020a. doi: 10.1162/tacl_a_00342. URL https://aclanthology.org/2020.tacl-1.48.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In *34th Conference on Neural Information Processing Systems 33*, 2020b. URL https://proceedings.neurips.cc/paper/2020/file/e992111e4ab9985366e806733383bd8c-Paper.pdf.

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot, and Louise Naudin. Pre-training is (almost) all you need: An application to commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3878–3887, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.357. URL https://www.aclweb.org/anthology/2020.acl-main.357.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015. URL https://dl.acm.org/doi/pdf/10.1145/2736277.2741093.

Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *Proceedings of the 25th international conference on world wide web*, pages 1419–1428, 2016. URL https://dl.acm.org/doi/pdf/10.1145/2872427.2874809.

Wilson L Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953. URL https://journals.sagepub.com/doi/pdf/10.1177/107769905303000401.

Jaime Teevan, William Jones, and Benjamin B Bederson. Personal information management. *Communications of the ACM*, 49(1):40–43, 2006. URL http://www.cs.umd.edu/~bederson/images/pubs_pdfs/p40-teevan.pdf.

James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. From natural language processing to neural databases. *Proceedings of the VLDB Endowment*, 14(6):1033–1039, 2020. URL http://vldb.org/pvldb/vol14/p1033-thorne.pdf.

James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. Database reasoning over text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3091–3104, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.241. URL https://aclanthology.org/2021.acl-long.241.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-4007. URL https://aclanthology.org/W15-4007.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1174. URL https://aclanthology.org/D15-1174.

Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1434–1444, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1136. URL https://aclanthology.org/P16-1136.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2016. URL http://proceedings.mlr.press/v48/trouillon16.pdf.

Théo Trouillon, Éric Gaussier, Christopher R Dance, and Guillaume Bouchard. On inductive abilities of latent factor models for relational learning. *JAIR*, 64, 2019. URL https://www.jair.org/index.php/jair/article/download/11305/26465/.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 2020a. URL https://ojs.aaai.org/index.php/AAAI/article/view/5694/5550.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/pdf?id=BylA_C4tPr.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/pdf?id=rJXMpikCZ.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W Cohen. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *arXiv preprint arXiv:2007.00849*, 2020. URL https://arxiv.org/pdf/2007.00849.pdf.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001. URL https://ieeexplore.ieee.org/iel5/7768/21353/00990517.pdf.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.670.9826&rep=rep1&type=pdf.

Mengting Wan and Julian McAuley. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM conference on recommender systems*, pages 86–94, 2018. URL https://dl.acm.org/doi/pdf/10.1145/3240323.3240369.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2605–2610, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1248. URL https://www.aclweb.org/anthology/P19-1248.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In

7th International Conference on Learning Representations, ICLR 2019, 2019a. URL https://openreview.net/pdf?id=rJ4km2R5t7.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1737–1748, 2021a. URL https://dl.acm.org/doi/pdf/10.1145/3442381.3450043.

Cunxiang Wang, Pai Liu, and Yue Zhang. Can generative pre-trained language models serve as knowledge bases for closed-book QA? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3241–3251, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.251. URL https://aclanthology.org/2021.acl-long.251.

Lidan Wang, Jimmy Lin, and Donald Metzler. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 105–114, 2011. URL https://dl.acm.org/doi/pdf/10.1145/2009916.2009934.

Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, et al. Deep graph library: Towards efficient and scalable deep learning on graphs. *arXiv preprint arXiv:1909.01315*, 2019b. URL https://arxiv.org/abs/1909.01315.

Quan Wang, Bin Wang, and Li Guo. Knowledge base completion using embeddings and rules. In *Twenty-fourth international joint conference on artificial intelligence*, 2015. URL https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/download/10798/10921.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29 (12):2724–2743, 2017. URL https://ieeexplore.ieee.org/iel7/69/4358933/08047276.pdf.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online, August 2021c. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.121. URL https://aclanthology.org/2021.findings-acl.121.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019c. URL https://dl.acm.org/doi/pdf/10.1145/3331184.3331267.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019d. URL https://dl.acm.org/doi/pdf/10.1145/3308558.3313562.

Xiaofang Wang, Dan Kondratyuk, Eric Christiansen, Kris M Kitani, Yair Alon, and Elad Eban. Wisdom of committees: An overlooked approach to faster and more accurate models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/pdf?id=MvO2t0vbs4-.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021d. doi: 10.1162/tacl_a_00360. URL https://aclanthology.org/2021.tacl-1.11.

Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 115–124. ACM, 2016. URL https://dl.acm.org/doi/pdf/10.1145/2911451.2911537.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014. URL https://ojs.aaai.org/index.php/AAAI/article/download/8870/8729.

Larry Wasserman. All of statistics: A concise course in statistical inference, 2010.

Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Found. Trends Databases*, 10 (2-4):108–490, 2021. doi: 10.1561/1900000064. URL https://www.nowpublishers.com/article/DownloadSummary/DBS-064.

Steve Whittaker and Candace Sidner. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 276–283. ACM, 1996. URL https://dl.acm.org/doi/fullHtml/10.1145/238386.238530.

Steve Whittaker, Tara Matthews, Julian Cerruti, Hernan Badenes, and John Tang. Am i wasting my time organizing email?: a study of email refinding. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 3449–3458. ACM, 2011. URL https://dl.acm.org/doi/pdf/10.1145/1978942.1979457.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association

for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019. URL http://proceedings.mlr.press/v97/wu19e/wu19e.pdf.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.519. URL https://www.aclweb.org/anthology/2020.emnlp-main.519.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020b. URL https://ieeexplore.ieee.org/iel7/5962385/9312808/09046288.pdf.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. From one point to a manifold: knowledge graph embedding for precise link prediction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016a. URL https://www.ijcai.org/Proceedings/16/Papers/190.pdf.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. TransG : A generative model for knowledge graph embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2316–2325, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1219. URL https://aclanthology.org/P16-1219.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016. URL https://www.ijcai.org/Proceedings/16/Papers/421.pdf.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/pdf?id=zeFrfgyZln.

Wenhan Xiong, Thien Hoang, and William Yang Wang. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1060. URL https://aclanthology.org/D17-1060.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/pdf?id=BJlzm64tDH.

Canran Xu and Ruijiang Li. Relation embedding with dihedral group in knowledge graph. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 263–272, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1026. URL https://aclanthology.org/P19-1026.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pages 5453–5462. PMLR, 2018. URL http://proceedings.mlr.press/v80/xu18c/xu18c.pdf.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/pdf?id=ryGs6iA5Km.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.523. URL https://www.aclweb.org/anthology/2020.emnlp-main.523.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462*, 2021. URL https://arxiv.org/pdf/2102.06462.pdf.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6575.

Longqi Yang, Tobias Schnabel, Paul N Bennett, and Susan Dumais. Local factor models for large-scale inductive recommendation. In *Fifteenth ACM Conference on Recommender Systems*, pages 252–262, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3460231.3474276.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016. URL http://proceedings.mlr.press/v48/yanga16.pdf.

Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019. URL https://arxiv.org/pdf/1909.03193.

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *arXiv preprint arXiv:1908.06725*, 2020. URL https://arxiv.org/pdf/1908.06725.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018. URL https://dl.acm.org/doi/pdf/10.1145/3219819.3219890.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*, 2020. URL https://arxiv.org/pdf/2010.00796.

Hamed Zamani, Michael Bendersky, Xuanhui Wang, and Mingyang Zhang. Situational context for ranking in personal search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1531–1540, 2017. URL https://dl.acm.org/doi/pdf/10.1145/3038912.3052648.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, volume 32, 2019a. URL https://proceedings.neurips.cc/paper/2019/file/d961e9f236177d65d21100592edb0769-Paper.pdf.

Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping Shi, Hui Xiong, and Qing He. Relational graph neural network with hierarchical attention for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6508/6364.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://www.aclweb.org/anthology/P19-1139.

Qian Zhao, Paul N Bennett, Adam Fourney, Anne Loomis Thompson, Shane Williams, Adam D Troy, and Susan T Dumais. Calendar-aware proactive email recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 655–664. ACM, 2018. URL https://dl.acm.org/doi/pdf/10.1145/3209978.3210001.

Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.398. URL https://aclanthology.org/2021.naacl-main.398.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020a. URL https://www.sciencedirect.com/science/article/pii/S2666651021000012.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9733–9740, 2020b. URL https://ojs.aaai.org/index.php/AAAI/article/view/6523/6379.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/58ae23d878a47004366189884c2f8440-Paper.pdf.

Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11168–11176, 2021. URL https://www.aaai.org/AAAI21Papers/AAAI-8484.ZhuJ.pdf.

Xiaojin Zhu. *Semi-supervised Learning with Graphs*. PhD thesis, 2005. URL http://pages.cs.wisc.edu/~jerryzhu/machineteaching/pub/thesis.pdf.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University*, 2002. URL https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.3864&rep=rep1&type=pdf.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015. URL https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Zhu_Aligning_Books_and_ICCV_2015_paper.pdf.