USER LATENT BEHAVIOR MODELING IN AN INTERCONNECTED WORLD

KANIKA NARANG

Dissertation
Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois
May 2020

Doctoral Committee:
Associate Professor Hari Sundaram, Chair,
Assistant Professor Alexander Schwing,
Professor ChengXiang Zhai,
Dr. Chris Brew

Ohana means family.
Family means nobody gets left behind, or forgotten.

— Lilo & Stitch

Dedicated to the loving memory of Rudolf Miede.
1939 – 2005

ABSTRACT

User behavior modeling has become an indispensable tool with the proliferation of socio-technical systems to provide a highly personalized experience to the users. These socio-technical systems are used in sectors as diverse as education, health, law to e-commerce, and social media. The two main challenges for user behavioral modeling are building an in-depth understanding of online user behavior and using advanced computational techniques to capture behavioral uncertainties accurately. This thesis addresses both these challenges by developing interpretable models that aid in understanding user behavior at scale and by developing sophisticated models that perform accurate modeling of user behavior.

Specifically, we first propose two distinct interpretable approaches to understand explicit and latent user behavioral characteristics. Firstly, in **??**, we propose an interpretable Gaussian Hidden Markov Model-based cluster model leveraging user activity data to identify users with similar patterns of behavioral evolution. We apply our approach to identify researchers with similar patterns of research interests evolution. We further show the utility of our interpretable framework to identify differences in gender distribution and the value of awarded grants among the identified archetypes. We also demonstrate generality of our approach by applying on StackExchange to identify users with a similar change in usage patterns.

Next in **??**, we estimate user latent behavioral characteristics by leveraging user-generated content (questions or answers) in Community Question Answering (CQA) platforms. In particular, we estimate the latent aspect-based reliability representations of users in the forum to infer the trustworthiness of their answers. We also simultaneously learn the semantic meaning of their answers through text representations. We empirically show that the estimated behavioral representations can accurately identify topical experts.

We further propose to improve current behavioral models by modeling explicit and implicit user-to-user influence on user behavior. To this end, in **??**, we propose a novel attention-based approach to incorporate influence from both user's social connections and other similar users on their preferences in recommender systems. Additionally, we also incorporate implicit influence in the item space by considering frequently co-occurring and similar feature items. Our modular approach captures the different influences efficiently and later fuses them in an interpretable manner. Extensive

experiments show that incorporating user-to-user influence outperforms approaches relying on solely user data.

User behavior remains broadly consistent across the platform. Thus, incorporating user behavioral information can be beneficial to estimate the characteristics of user-generated content. To verify it, in Chapter 3, we focus on the task of best answer selection in CQA forums that traditionally only considers textual features. We induce multiple connections between user-generated content, i.e., answers, based on the similarity and contrast in the behavior of authoring users in the platform. These induced connections enable information sharing between connected answers and, consequently, aid in estimating the quality of the answer. We also develop convolution operators to encode these semantically different graphs and later merge them using boosting.

We also proposed an alternative approach to incorporate user behavioral information by jointly estimating the latent behavioral representations of user with text representations in **??**. We evaluate our approach on the offensive language prediction task on Twitter. Specially, we learn an improved text representation by leveraging syntactic dependencies between the words in the tweet. We also estimate the abusive behavior of users, i.e., their likelihood of posting offensive content online from their tweets. We further show that combining the textual and user behavioral features can outperform the sophisticated textual baselines.

PUBLICATIONS

Some ideas and figures have appeared in the following publications:

Morales, Alex, Kanika Narang, Chengxiang Zhai, and Hari Sundaram (2020). “CrowdQM: Learning User Aspect-Reliability and Comment Trustworthiness in Discussion Forums.” In: *24th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

Narang, Kanika and Chris Brew (2020). “Hate Speech Classification using Syntactic Dependency Graphs.” In: *To be submitted*.

Narang, Kanika, Austin Chung, Hari Sundaram, and Snigdha Chaturvedi (2019a). “Discovering Archetypes to Interpret Evolution of Individual Behavior.” In: *arXiv preprint arXiv:1902.05567*.

– (2019b). “Discovering Archetypes to Interpret Evolution of Individual Behavior.” In: *arXiv preprint arXiv:1902.05567*.

Narang, Kanika, Susan T. Dumais, Nick Craswell, Dan Liebling, and Qingyao Ai (2017). “Large-Scale Analysis of Email Search and Organizational Strategies.” In: *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. New York, NY, USA: ACM. ISBN: 978-1-4503-4677-1. DOI: 10.1145/3020165.3020175.

Narang, Kanika, Yitong Song, Alexander Schwing, and Hari Sundaram (2020). “FuseRec: Fusing user and item homophily modeling with temporal recommender systems.” In: *Under review at ECML-PKDD*.

Narang, Kanika, Chaoqi Yang, Adit Krishnan, Junting Wang, Hari Sundaram, and Carolyn Sutter (2019). “An Induced Multi-Relational Framework for Answer Selection in Community Question Answer Platforms.” In: *arXiv preprint arXiv:1911.06957*.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth (Knuth, 1974)

ACKNOWLEDGMENTS

This thesis has been made possible due to the contributions of many people. First and foremost, I would like to express the sincerest gratitude to my advisor, Associate Professor Hari Sundaram, for his constant support and guidance. My most prominent learning from his mentorship is his unique problem-solving skills and critical thinking. His attention to detail, including presentation skills, has also improved clarity in my academic writing. With the independence afforded by him in my research, I was able to explore and work in diverse areas helping me become a more holistic researcher.

I also extend my heartfelt thanks to Assistant Professor Alexander Schwing for taking me under his wings and teaching me the ropes of deep learning. I appreciate him devoting so much time to mentor me even when our research interests did not align. I thoroughly enjoyed working with him, and he is among the most optimistic and supportive people I know in academia. Next, I would like to thank my committee members, Professor ChengXiang Zhai and Dr. Chris Brew, for their contributions to this thesis. Prof. Zhai has taught me the importance of situating your research in the grand scheme of things. He has pushed me harder to think about the broader impact of my research that has dramatically improved the writing quality of this thesis. I also thank Dr. Brew for teaching me the importance of fundamental research and to being critical even of positive results.

I am also grateful to Assistant Professor Snigdha Chaturvedi and Dr. Susan Dumais for mentoring me in the initial years of my Ph.D. and providing an exemplar example of successful women researchers in the field. I feel fortunate to be advised by exceptional mentors in my career. A big shout out to especially Professor Pankaj Jalote, Professor Ponnurangam Kumaraguru (PK), and Professor Mayank Vatsa at my undergraduate institution, Indraprastha Institute of Information Technology (IIIT-D), for igniting the flame of research curiosity in my formative years. My initial

experiences during IIIT-D helped me appreciate the joys and importance of research and motivated me to pursue it as a full-time career.

Thanks also to my colleagues at Crowd Dynamics Lab for being my mentors, support system, and critiques of my research. All my friends outside work also deserve a special mention for helping me maintain a semblance of a personal life beyond my research.

Finally, to my parents for believing in my dream and their continued love and support over the years. This acknowledgment will not be complete without mentioning my fiancé, Dr. Ketan Mittal. Thanks for being my pillar of strength and an ardent champion of my success throughout this time. I owe a great deal of this thesis to him.

CONTENTS

III APPENDIX

1 INTRODUCTION

Example is the school of mankind,
and they will learn at no other.
—Edmund Burke

Artificial Intelligence (AI) is a branch of computer science research that deals with the development of machines with intelligence rivaling those of humans. In other words, machines that can perform tasks that generally require human intelligence, such as visual perception, decision-making, speech recognition, and more. Most of the current AI research is task-driven and has achieved or, in some cases, even surpassed human intelligence (Devlin et al., 2019; K. He et al., 2015; Silver et al., 2016; Szegedy et al., 2015). This advancement has resulted in the assimilation of AI into our life inadvertently, in the form of *socio-technical* systems around us. These intelligent systems provide self-paced learning in the education sector (Jiang et al., 2015), enable targeted marketing in e-commerce websites (Yager, 2000), facilitate personalized medicare unique to patient's body type, genetics, and lifestyle (La Thangue and Kerr, 2011) and even predict repeat crime incidence for bail seeking convicts (Kleinberg et al., 2018).

The success of these intelligent machines lies in the fact that they can understand and model complex human behavior effectively and use it to extend highly personalized solutions at scale. Interactions or activities performed by the user on a specific platform characterize their behavior. For instance, in an e-commerce platform, activities denote items purchased by the user. Similarly, in a Community Question Answering (CQA) forum like StackExchange or Reddit, these activities are defined as posting questions or answers or voting on other user's answers. Artificial Intelligence, or specifically user behavior modeling, sifts through vast amounts of past user data to find recurring patterns and predict user's future purchases, search intent, or information need (Agichtein et al., 2006; Beutel, 2016; Kang and McAuley, 2018).

There are still many challenges abound to accurate modeling of user behavior. These challenges primarily arise because humans are imperfect sensors of information. They do not necessarily conform to repetitive patterns and can be unpredictable. The user's behavior also tends to evolve. Besides, users are biased as they tend to get influenced by other users, their environment, or even exhibit unconscious biases. With that being said, the current scenario also affords many more opportunities that were not present before. First, due to the close intertwining of the technology with our lifestyle, we have abundant user interaction data available to us now, more than ever before. These vast reserves of longitudinal data can help the models to learn the nuances in the user behavior with more traces of behavioral change over time. Second, they are further aided by the concurrent improvements in computing power and computational techniques, such as deep neural networks, that can learn higher-degree polynomial functions needed to model and understand complex user behavior.

The philosophy of achieving at par human intelligence through AI is, in essence, to first understand how humans process and extract knowledge from their environment and then emulate that in machines. Thus, to build a comprehensive user behavior model, we need to both *understand* their online behavior and use the learned insights to accurately *model* the user activity data in the platform.

Recently proposed Deep Neural Networks (DNNs) are the class of AI algorithms that employ multiple layers to progressively extract more abstract and composite representations from the raw input, thus, removing the need for feature engineering. DNNs have also beaten human benchmarks in many language understanding and visual perception tasks (Gilbert, 2013b; Szegedy et al., 2015), but they are notorious for being uninterpretable. The model interpretability is highly desirable as these models are used increasingly in sensitive and impactful domains like law and health. Owing to the demand for explainable models, a recent class of works advocates using simple *interpretable* models such as decision trees, linear regression for these domains (Lakkaraju and Rudin, 2017). On the other hand, model-agnostic approaches are also developed for interpreting these black-box neural models with techniques like feature importance or explaining individual predictions (**lime**; Lakkaraju, Kamar, et al., 2017). However, it is still challenging to achieve the dual objective of *interpretability* and high *precision* in a single model. It thus creates a dichotomy between creating simpler models that offer a more in-depth understanding of the behavior versus using the advanced computational techniques to capture behavioral uncertainties accurately.

1.1 USER BEHAVIOR MODELING

The overall aim of this dissertation is to achieve both *understanding* and an accurate *modeling* of user behavior. However, it is not easy to build *interpretable* models that aid in understanding users and are also *sophisticated* enough to capture behavior nuances precisely. Thus, we propose to view the problem of user behavior modeling from multiple angles. Each perspective will lead us to a different class of solutions, all of them bringing us closer to the overall goal of improved user behavior modeling.

1.1.1 *Understanding user behavior*

An abundance of user activity data online presents tremendous opportunities to analyze user behavior unhindered in the real world. This activity data also often spans thousands or millions of users; a scale never achievable in field experiments. This opportunity has led to the emergence of an interdisciplinary field known as *computational social science* that brought social scientists and computer scientists together. Researchers in this field use computational techniques to investigate behavioral relationships and social interactions in online platforms (Wagner et al., 2016; X. Zhou et al., 2018). They typically assess the validity of previous social science theories to understand user online behavior at scale.

The first and foremost perspective, thus, draws from the field of computational social science. The works following this perspective should use advances in computational techniques to process vast amounts of user data and extract meaningful and comprehensible patterns of user behavior. These models, primarily aiming at providing an in-depth understanding of user behavior, thus fall into the category of *interpretable* models. This interpretability often comes at the expense of model precision.

These models can also provide an excellent framework to perform additional hypothesis testing of correlation of user's behavior with other covariates that can be possibly predictive of their behavior. For instance, we can empirically test hypotheses like do changes in the posting pattern of a StackExchange user affect the upvotes their answers get? Furthermore, does that subsequently affect their activity level in the platform? These findings can be beneficial for moderators of the online communities to devise incentivization strategies to retain active users in the platform. Another interesting hypothesis worth investigating can be the correlation of change in the publication behavior of scholars with the amount of research grants awarded to them. These findings can be particularly attractive to grant-awarding institutions to ascertain any potential biases or merits in their current grant-awarding scheme.

Apart from discrete user activity data, there are massive amounts of multimodal user interaction data available in these platforms such as text, video, speech, etc. User-generated textual data is the most popular form of interaction among them. Textual data is prevalent on multiple platforms such as reviews in e-commerce websites, questions, or answers text on CQA forums, tweets, or posts on social media platforms like Twitter or Facebook.

Textual data is more complex and sophisticated to comprehend than discrete activity features. Nevertheless, analyzing textual data opens the door to *understanding* the extensive and latent characteristics of user behavior that are not even possible to comprehend with activity features. For instance, the text of user reviews can be used to learn user affinity to different aspects of the product, such as relative importance of different aspects-food, service, location of a restaurant for a particular user. Similarly, the text of the user's answers or questions in a CQA forum can be used to discern latent features like the user's preferable topics to answer or their expertise for different topics. Prior works have leveraged the text of user's tweets or posts to understand user's political leanings (Wong et al., 2016), state of their mental health (De Choudhury et al., 2016), and much more. Thus, it is imperative to leverage user-generated content to create a comprehensive understanding of user behavior.

Finally, works following this line of research should build interpretable models using extensive user data to provide an in-depth understanding of user behavior at scale.

1.1.2 *Improving user behavioral models*

The technological advancement in precise modeling of user behavior has afforded us with the seamless integration of AI into our lives. These intelligent systems provide *personalized* solutions at scale in sectors as diverse as e-commerce to education to medicine. Another complementary usage of creating powerful behavioral models is the ability to identify deviant users or even credible users on the platform. This outcome is highly desirable in the current circumstances, with the rise in illegitimate use of technology.

The second perspective, thus, mainly deals with pushing the performance boundaries of the state-of-the-art user behavioral models. Specifically, the solutions following this line of research strive to capture a comprehensive picture of user behavior by factoring in the myriad explicit and implicit influences on users. These methods can draw from a large body of social science research about user behavior in real-life settings. Note that the first perspective also deals with evaluating these theories in the online data traces with the primary aim of interpretability. As noted earlier, interpretability often occurs at the cost of model precision. On the contrary, the primary aim of this line of research is to build sophisticated behavioral models that can predict future user behavior accurately. In fact, the second perspective follows from the first one as it can leverage an improved understanding of the user's online behavior to build precise behavioral models.

The works following this line of research need to tackle multiple challenges posed to accurate behavioral modeling. For instance, a user's behavior tends to evolve with experience. Similarly, a user's friends, peers, or in general, other users with a similar background (demography, preferences, etc.) often influence their behavior. It is now possible to computationally model these effects due to the availability of extensive user interaction data on the online platforms. For instance, long-term user data provides the opportunity to model patterns of change in user behavior with time. Similarly, user-to-user influences manifest in many of the current online platforms due to their prevalent social structures. Connected users, i.e., users with established trust or friend relationships on these platforms, are empirically shown to exhibit similar behavior online, a phenomenon popularly known as user homophily (Jie Tang et al., 2009).

Further, users hold an unconscious bias towards users with a similar background; a phenomenon also reverberated online. For instance, a user may trust movie recommendations of another user with similar demographics (same age or gender). Similarly, an Indian user may trust the ratings of another Indian user more than a non-Indian user when evaluating an Indian restaurant. Thus, it is crucial to capture these implicit influences between users who are alike based on general notions of similarity, such as demography or activity in the platform. Capturing the implicit influence is more complicated than explicit influence, but it can provide vital cues for predicting the behavior of users with *few* social connections. They can also be particularly helpful in online platforms with no established social structure such as review platforms or CQA forums.

Thus, the creation of models that capture the explicit and implicit influences on users efficiently and at scale is prudent to bring advancement in the field of user behavior modeling.

1.1.3 *Incorporating user behavior as metadata to improve complementary tasks*

Most of the current research related to user behavior modeling intends to model or predict user behavior primarily. However, there are related tasks pertinent to the estimation of characteristics of the user-generated content in the online platforms. Some of the examples of such related tasks are estimating the quality, credibility, or profanity of the content online.

Current work solving these tasks merely exploits the data features and completely ignores the user information. Users are the creators of the content, and their behavior remains broadly consistent across the platform. Thus, adding contextual information about user behavior estimated from their actions in the platform can immensely improve the prediction task. For instance, current models proposed for prediction tasks like credible answer selection on CQA forums or hate speech prediction utilize the semantic meaning of the text to make such predictions (ElSherief et al., 2018; H. Zhang et al., 2018). However, user expertise estimated through their prior answers on the platform can be used to differentiate between users. This differentiation can consequently help to rank user-provided answers based on their trustworthiness. Similarly, the abusive behavior of users estimated from their prior content can provide a useful precedent when predicting the offensive nature of their new content. Furthermore, utilizing user homophily, behavioral priors of users can be shared amongst explicitly and implicitly similar users.

Thus, in this perspective, we propose to build models that leverage information about commonalities and disparities in user behavior to improve prediction tasks about user-generated content.

1.2 THESIS CONTRIBUTIONS

We outlined three different perspectives to solve the problem of user behavior modeling. These different perspectives are in no means comprehensive. However, they pave a viable way to ultimately develop models than can attain the twin goal of interpretability and precision. Since the field of user behavioral modeling is massive, there are numerous unsolved challenges within each perspective. In this dissertation, we propose a few foundational works under each perspective that provide potential approaches to solve these posed challenges. Specifically, we attempt to answer the dichotomy of understanding versus modeling user behavior by proposing interpretable models that primarily aim to provide detailed insights about user online behavior. Further, we develop frameworks to model user behavior accurately or leverage information about user behavior to improve prediction tasks related to user-generated content.

1.2.1 *Understanding user behavior*

Under this perspective, we propose two works, the first one that directly models user activity data to understand patterns of behavioral change and another that leverages user-generated content to understand the latent characteristics of user behavior.

Firstly, in Chapter **??**, we leverage user activity data to understand the behavioral evolution of individuals with experience. We introduce an interpretable Gaussian Hidden Markov Model (G-HMM) cluster model to identify archetypes of evolutionary patterns among users. Specifically, we apply our model to discover archetypical patterns of research interests' evolution among Academics and patterns of change in activity distribution of users of Stack Exchange communities. Our model allows us to correlate user behavior with external variables such as gender, income, etc.

In **??**, we leverage the content of the user's answers in CQA forums to learn latent characteristics of user behavior–*latent reliability*. We use this latent behavior representation to solve the task of ranking answers of a given question based on its trustworthiness. This ranking is especially vital as CQA forums are crippled with rampant unreliable content on their platform due to almost no regulations on post requirements or user background. Thus, this misinformation severely limits the forum's usefulness to its users.

We propose an unsupervised framework to learn the latent characteristic of user behavior–reliability and latent characteristic of answers–trustworthiness in a mutually reinforcing manner. In particular, our model learns a user representation vector capturing her reliability over fine-grained topics discussed in the forum. Besides, we also learn the semantic meaning of comments and posts through text-aware text representations or word embeddings. The learned latent representations using text affords an in-depth understanding of user reliability, improbable to comprehend using discrete activity data.

1.2.2 *Improving user behavioral models*

There are multiple unsolved challenges for accurate modeling of user behavior online. In this dissertation, we focus on capturing the user-to-user influence to improve user behavioral models. These influences can be either explicit in terms of social connections present in the platform itself or implicit in the absence or sparsity of established social connections. We propose to capture the implicit social influence, measured either through similarity or contrast in users' behaviors, by inducing connections between them. These connections enable information sharing among connected users resulting in an improved model of their behavior.

We use *Graph Convolution Networks* (GCN) (Kipf and Welling, 2016a) to model both explicit social connections and induced connections between users. GCN is a recent class of neural networks that learns node representations in graph-structured data. Specifically, the model aggregates representations of the node itself, along with its neighbors, to compute a node representation. The model is very efficient with parallel batch processing and sparse computations. Thus, it can scale to large scale user graphs present on online platforms.

Recommender Systems have previously exploited the user homophily (similar behavior) between connected users to provide improved recommendations to their users (Jannach and Ludewig, 2017; Le Wu et al., 2019; Zhao, McAuley, and King, 2014). Thus, in Chapter **??**, we propose to incorporate the effect of *user-to-user influence* on the user's behavior in a recommender system. In this work, we exploit homophily in both user and item space. In the user space, apart from a user's explicit social connections in the platform, we also induce connections between users with a similar purchasing history. In the item space, we construct a 'social graph of items' based on similarity in item features and co-occurrence in the dataset. These implicit similarity connections between items help the model to handle data sparsity in items (long-tail items, i.e., items with limited training data).

We propose a novel graph attention-based aggregation models to estimate social influence in both user social and item similarity graphs. Besides, we also learn explicit attention weights for each pair of connected nodes to capture varying influence strengths on the behavior. We finally propose an interpretable aggregation approach to combine the different factors influencing user preferences.

1.2.3 *Incorporating user behavior as metadata to improve complementary tasks*

Under this perspective, we leverage user behavior information to aid in two diverse prediction tasks related to user-generated content. In addition, we propose distinct techniques to include user behavioral information for each task. The first approach incorporates the user behavioral information by inducing connections amongst user-generated content. Induced connections aid in information sharing resulting in improved predictions. The second technique, on the other hand, learns powerful user representations encapsulating users' behavior. We subsequently use these representations in addition to the textual features to improve the prediction task.

CQA forums suffer from abundant low quality content and answer selection task, thus, aims at identifying the best answer out of the given answers to a question. Current approaches predict answer quality in isolation of the other answers to the question and user activity across the forum (other posted questions or answers). Thus, in Chapter 3, we induce connections based on both similarity and contrast between users' behavior (answers) to share user behavioral information among answers.

Specifically, we induce a contrastive graph between user-provided answers replying to the same question and a similarity graph between answers across different questions if the replying users are exhibiting similar behavior. We also propose a modification to the original GCN to encode the notion of contrast between a node and its neighborhood. Besides, we use state-of-the-art text representation learning approaches to compute representation for the user's answers and questions. We subsequently induce connections between user-generated answers based on these text representations. Finally, multiple graphs expressing semantically diverse relationships are merged through an efficient boosting architecture to predict the best answer.

Thereafter, in **??**, we work on leveraging textual features along with user features to detect the offensive language in tweets. Abusive behavior is rampant online and is affecting the experience of a large number of users on the platform. Hate attacks are often expressed in a sophisticated manner in the text (long clauses or complex scoping); thus, traditional sequential neural models are unable to capture them effectively. In this work, we learn an improved text representation of the tweets by leveraging syntactic dependencies between words. We achieve this by inducing a graph on the words of a tweet where edges represent a dependency relationship. We use these representations subsequently to estimate a user's latent abusive behavior, i.e., their likelihood of using offensive language online. Further, to capture homophily in abusive user accounts, we propagate this latent behavior through the user's social graph on Twitter. This user behavior information, in addition to the improved text representation of the tweet, dramatically improves the performance of offensive language detection models.

Part I

FOUNDATIONS

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2 LITERATURE REVIEW

Before delving into details of our proposed approach, we first discuss prior literature related to User Behavior Modeling in Online Social Networks, Academic Dataset and Recommender Systems. We also review text representation approaches used in Community Question Answering (CQA) forums and for short text in Twitter. We also briefly review recently proposed Graph convolution networks to model graph-structured data (used in our work) for these platforms.

2.1 ONLINE SOCIAL NETWORKS

There has been a lot of interest in the past on identifying and characterizing user behavior in online social networks (OSNs). Maia, J. Almeida, and V. Almeida (2008) identified five distinct user behaviors of YouTube users based on their individual and social attributes. While Mamykina et al. (2011) identified user roles based on just answer frequency in StackExchange. Adamic et al. (2008) and Furtado et al. (2013) worked on similar user behavioral studies on Yahoo Answers and Stack Overflow datasets, respectively. All these studies, however, ignore *temporal changes* in the behavior and use engineered features for behavior modeling.

Some behavioral studies do model evolution of user activities in the platform too. Benevenuto et al. (2009) learned a Markov model to examine transition behavior of users between different activities in Orkut in a static snapshot. J. Yang et al. (2014) and Knab et al. (2003) proposed generative models that assigned each user action to a progression stage and classify event sequences simultaneously. They used their model to predict cancer symptoms, or products user would review in the future. However, the model did little to provide meaningful and interpretable stages and clusters. Angeletou, Rowe, and Alani (2011) constructed handcrafted rules to identify user roles and studied the change of user roles' composition in the community over time. Recently, Santos et al. (2019) identified four distinct types of user activity pattern based on their activity frequency.

The Hidden Markov Model (HMM) has been widely used to model and cluster time sequences (Bicego, Murino, and Figueiredo, 2003; Coviello, Chan, and Lanckriet, 2014; Smyth, 1997) in the past. However, most of these models learn an HMM for each user sequence and then employ clustering algorithms to cluster the learned HMMs. These approaches are not scalable, and the clusters thus identified are not interpretable.

2.2 RECOMMENDER SYSTEMS

In the following section, we provide a brief review of approaches that model users' historical interactions to improve recommender systems. We first enlist approaches assuming static user behavior (Collaborative Filtering). Consequently, we review approaches that model the evolution of user behavior (Temporal Recommendation), social influence (Social Recommendation), and few recently proposed methods which are looking at combining the two (Socio-Temporal Recommendation).
Collaborative Filtering: Collaborative Filtering (CF) is one of the most popular techniques for user modeling in recommender systems. Specifically, the methods employ Matrix Factorization (MF) to decompose a user-item rating matrix into user and item specific latent factors. Classical and seminal work for MF-based recommender systems (Rendle, Freudenthaler, Gantner, et al., 2009) uses a Bayesian pairwise loss (BPR). Collaborative filtering is also performed in item space (Sarwar et al., 2001), where similar items are computed offline based on their rating similarity or co-occurrence in the dataset. Consequently, it recommends items similar to the ones used in the past by the user. Neural net approaches have been proposed recently to improve MF models. They learn more complex non-linearities in the user-item interaction data (X. He et al., 2017; Yao Wu et al., 2016).

However, most MF approaches assume a static user-item interaction matrix. Often, this assumption is not accurate, particularly for online communities where user preferences evolve over time — sometimes quickly — necessitating temporal recommendation.
Temporal Recommendation: There has been significant work in the area of temporal recommender systems that model a user's past interactions to inform a user's current preference. These temporal models generally assume a linear relationship between the events and model it using a Markov chain (Cheng et al., 2013; Rendle, Freudenthaler, and Schmidt-Thieme, 2010). However, these are often 'shallow' (i.e., linear) methods that are inept at modeling the more complex dynamics of temporal changes. Recent works (Cai, R. He, and McAuley, 2017; Kang and McAuley, 2018; P. Sun, Le Wu, and Meng Wang, 2018) use deep net approaches involving convolution layers, attention networks, and recurrent neural nets to model complex relations. For example, Jiaxi Tang and K. Wang (2018) applies

convolutional filters on the embedding matrix computed from a few recent items of a user. This model captures a higher-order Markov chain, but it still has a limited scope as it does not consider the entire history of a user. In contrast, to model long term dependencies, Jannach and Ludewig (2017) propose to model a user's sequential behavior within a session using recurrent neural nets. C.-Y. Wu et al. (2017) apply a recurrent architecture to both user and item sequences and hence model dynamic influences in popularity of movies on users' viewing preference. Kang and McAuley (2018) instead employ a self attention module for next item recommendation that adaptively learns the importance of all past items in a user's history. However, these models are limited as they do not leverage the social connections of a user.

Social Recommendation: Social recommenders integrate information from a user's social connections to mitigate data sparsity for cold-start users, i.e., users with no or minimal history. They exploit the principle of social influence theory (Jie Tang et al., 2009), which states that socially connected users exert influence on each other's behavior, leading to a homophily effect: similar preferences towards items. Jamali and Ester (2010) and H. Ma et al. (2011) use social regularization in matrix factorization models to constrain socially connected users to have similar preferences. The recently proposed SERec (Menghan Wang et al., 2017) embeds items seen by the user's social neighbors as a prior in an matrix factorization model. The SBPR model (Zhao, McAuley, and King, 2014) extends the pair-wise BPR model to incorporate social signals so that users assign higher ratings to items preferred by their friends. However, these models assume equal influence among all social neighbors. TBPR (X. Wang et al., 2016) distinguishes between strong and weak ties only when computing social influence strength.

Socio-Temporal Recommendation: Few of the recent approaches have started to look at merging temporal dependence with social influence. Cai, R. He, and McAuley (2017) extend Markov chain based temporal recommenders (Rendle, Freudenthaler, and Schmidt-Thieme, 2010) by incorporating information about the last interacted item of a user's friends. This work assumes markov dependence i.e. the future item just depends on the current item. This assumption is limiting im modeling evolving user preferences.

In the context of session-based recommendation, P. Sun, Le Wu, and Meng Wang (2018) propose a socially aware recurrent neural network that uses a dynamic attention network to capture social influence. On the other hand, W. Song et al. (2019) use graph attention nets to model social influence on a user's behavior in the session. Both these models learn a unified user representation based on social influence with a user's temporal history.

2.3 SCHOLARLY DATA

Most of the work on user behavioral mining concerns career movement within academia. Deville et al. (2014) observed that transitions between academic institutions are influenced by career stage and geographical proximity. While Clauset, Arbesman, and Daniel B Larremore (2015) found that academic prestige correlates with higher productivity and better faculty placement. Recently, Safavi, Davoodi, and Koutra (2018) studied career transitions across academia, government, and industry for Computer Science researchers. Dashun Wang, C. Song, and Barabási (2013) proposed a statistical model to predict the most impactful paper, in terms of citations, of scientists across disciplines. They argued nonexistence of a universal pattern and showed that highest-impact work in a scientist's career is randomly distributed within her body of work.

Recent studies also looked at gender differences in funding patterns, productivity, and collaboration trends in academia (Way, Daniel B. Larremore, and Clauset, 2016; Way, Morgan, et al., 2017). Way, Daniel B. Larremore, and Clauset (2016) did not observe any significant difference across gender in hiring outcomes in academia. However, they showed that indirect gender differences exist in terms of productivity, postdoctoral training rates, and in career growth. Some earlier studies also reported gender differences in academia. Kahn (1993) identified gendered barriers in obtaining tenure for academics in economics, while Ward (2001) found gendered differences in pay related to publication record.

There also has been considerable interest in mining scholarly data produced by researchers (bibliographic data, researchers' usage of social media, etc.). Prior studies have looked at the evolution of research interests on a community level. Liu et al. (2014) studied the evolution of research themes in articles published in CHI conference on Human Computer Interaction through co-word analysis. They highlighted specific topics as popular, core, or backbone research topics within the community. While Biryukov and Dong (2010) compared different scientific communities in DBLP dataset in terms of its interdisciplinary nature, publication rates, and collaboration trends. They also studied the variation of author's productivity with career length and observed that most of the authors have a short career spanning less than five years. Chakraborty and Nandi (2018) studied trajectories of successful papers in computer science and physics by analyzing paper citation counts. They classified these trajectories into multiple categories including early riser, a late riser, steady riser, and steady dropper.

2.4 COMMUNITY QUESTION ANSWERING FORUMS

Community Question Answering forums are increasingly used to seek advice online; however, they often contain conflicting and unreliable information. This misinformation could lead to serious consequences to the users. Thus, most of the work that model user behavior in CQA forums deals with predicting user reliability or quality of posted answers to a question.

Prior works can be classified into Feature-driven models; which use user and content-based engineered features for the task; another is Deep Text models that only model relevance of the content of question and answers for prediction and disregard user information. Recently, unsupervised approaches based on Truth Discovery principle are applied to model user expertise and answer quality simultaneously in these forums.

Feature-Driven Model: Feature-driven models (Burel, Mulholland, and Alani, 2016) develop features from three different perspectives: user features, content features, and thread features. These features are fed into classifiers, such as tree-based models (Burel, Mulholland, and Alani, 2016; Jenders, Krestel, and Naumann, 2016; Tian, P. Zhang, and B. Li, 2013) to identify the best answer. Tian, P. Zhang, and B. Li (2013) found that the best answer is usually the earlier and most different one, and tends to have more details and comments. Jenders, Krestel, and Naumann (2016) trained several classifiers for online MOOC forums. Different from existing works, Burel, Mulholland, and Alani (2016) emphasize on the thread-like structure of question & answer and introduce four thread-based normalization methods. These models predict the answer label independently of the other answers for the question. CQARank leverages voting information as well as user history and estimates user interests and expertise on different topics (L. Yang et al., 2013). Barrón-Cedeno et al. (2015) also look at the relationship between the answers, measuring textual and structural similarities between them to classify useful and relevant answers. All these supervised approaches need a large amount of labeled training data (Mihaylova, Nakov, et al., 2018; Oh, Yoon, and Kim, 2013; Wen et al., 2018). However, it is expensive and unsustainable to curate each answer manually for training these models. Alternatively, forums employ crowd sourced

voting mechanisms to estimate information reliability but it could lead to under-provision (Gilbert, 2013b).
Deep Text Models: Text-based deep learning models learn an optimal representation of question-answer text pairs suitable to select the best answer (Di Wang and Nyberg, 2015; W. Wu, H. Wang, and X. Sun, 2018; X. Zhang et al., 2017). In SemEval 2017 on Community Question Answering (CQA), (Nakov et al., 2017) developed a task to recommend useful related answers to a new question in the forum. SemEval 2019 further extends this line of work by proposing fact checking in community question answering (Mihaylova, Karadzhov, et al., 2019). Feng et al. (2015) augment CNN with discontinuous convolution for a better vector representation; Di Wang and Nyberg (2015) uses a stacked biLSTM to match question and answer semantics. Sukhbaatar et al. (2015) use attention mechanism in an end-to-end memory framework. Text-based models take longer to train and are computationally expensive.
Truth discovery: Different approaches based on truth discovery principle have been proposed to address predict answer quality in CQA forums (Q. Li et al., 2016; Y. Li, Q. Li, et al., 2015; Mukherjee et al., 2016; Vydiswaran, Zhai, and Roth, 2011; H. Zhang et al., 2018; Zheng et al., 2017). Many truth discovery approaches are tailored to categorical data and thus assume there is a single objective truth that can be derived from the claims of different sources (Y. Li, Gao, et al., 2016). Faitcrowd (F. Ma et al., 2015) assumes an objective truth in the answer set and uses a probabilistic generative model to perform fine-grained truth discovery. It jointly models the generation of questions and answers to estimate the source reliability and correct answer. On the other hand, Wan et al. (2016) propose trustworthy *opinion* discovery where the true value of an entity is modeled as a random variable with a probability density function instead of a single value.

Some truth discovery approaches also leverage text data to identify correct responses better. Y. Li, Du, et al. (2017) proposed a model for capturing semantic meanings of crowd provided diagnosis in a Chinese medical forum. In particular, they use a medical-related dictionary to extract terms in the response text and learn their semantic representations to discover trustworthy answers from non-expert users in crowdsourced diagnosis. H. Zhang et al. (2018) proposed a Bayesian approach to capture the multi-factorial property of text answers and used semantic representations of keywords to mitigate the diversity of words in answers. To model the user reliability, the authors proposed a two-fold reliability metric that uses both false positive and true positive rates. These approaches only use certain keywords for each answer and are thus, limited in their scope.

2.5 TWITTER

Most previous methods for detecting offensive speech on Twitter rely entirely on the textual content. Most of these prior work includes using statistical features like bag-of-words or tf-idf features for automated detection.Wulczyn, Thain, and Dixon (2017) used character n-gram features for detecting abusive comments in the discussion on Wikipedia pages. On Twitter dataset, Waseem and Hovy (2016) used character and word n-gram features along with lexical and users features to detect hate speech. Davidson et al. (2017) worked with character n-grams on a different Twitter dataset to achieve competitive performance. On the other hand, Nobata et al. (2016) combined n-grams features with linguistic, syntactic, and semantic features. However, they observed that n-gram features are most beneficial for the detection task. Even though bag-of-words approaches perform well, they are unable to capture nuanced hate speech as they fail to contextualize the word meanings. For instance, depending on the context, the word *gay* can be used to denote either ebullience or sexual preference. Only the latter is a candidate attack.

Recently, deep learning models are also proposed that leverage pre-trained word embeddings such as word2vec (Mikolov et al., 2013) and Glove (Pennington, Socher, and Manning, 2014) to capture aspects of the semantics of the tweets. These models aggregate individual word embeddings in a context-aware manner to compute tweet embeddings and later use them for classification. Gambäck and Sikdar (2017) and Park and Fung (2017) used the Convolutional Neural network to compute the tweet embeddings while Badjatiya et al. (2017) and Agrawal and Awekar (2018) showed that Gated Recurrent Units or Long-Short Term Memory networks are useful to compute these embeddings. On the other hand, Z. Zhang, Robinson, and Tepper (2018) used a combination of CNNs and GRU to achieve competitive performance.

The syntactic structure of the text can also be used to help identify the target group and the intensity of hate speech. For instance, Warner and Hirschberg (2012) extracts POS-based trigrams such as DT jewish NN to extract hate speech against a specific target, Jews. While, Silva et al. (2016) extends it further to look for generic syntactic structures like "I <intensity> hate <target>'. The primary difficulty of this work is that the space of possibly relevant rules is too large for an analyst to be confident that the list is truly comprehensive. In addition, it verges on the impossible to specify a set of rules that will do a decent job on the endless variety of possible implicit attacks.

A minority of approaches take advantage of non-textual user data in addition to the text. Pavlopoulos et al. (2017) added randomly-initialized user embeddings to their RNN model to obtain higher accuracy. Qian et al. (2018) showed that incorporating intra-user and reinforced inter-user representations significantly improve the performance of their bi-directional LSTM model. However, both of these approaches work on the individual user level and ignore the social influence on their behavior. Mishra et al. (2018) captured the social influence in abusive accounts by computing a representation of a user's neighborhood through node2vec features. The classifier described in Mishra et al. (2019) extends the previous paper by computing a user representation from an extended graph of users and tweets.

2.6 GRAPH CONVOLUTION NETWORKS

More recently, Graph Convolution Networks (GCNs) have been proposed to learn embeddings for graph-structured data (Kipf and Welling, 2016b). Graph Convolution can be applied in both spatial and spectral domains to compute node representations. The learned node representations are then used for various downstream tasks like node classification (Kipf and Welling, 2016a), link prediction (Schlichtkrull et al., 2018), multi-relational tasks (Sankar, Krishnan, et al., 2019) etc. Spatial approaches employ random walks or k-hop neighborhoods to compute node representations (Grover and Leskovec, 2016; Perozzi, Al-Rfou, and Skiena, 2014; Jian Tang et al., 2015; Z. Yang, Cohen, and Salakhutdinov, 2016). Pioneer works on graph convolution in the spectral domain use fast localized convolutions (Defferrard, Bresson, and Vandergheynst, 2016; Duvenaud et al., 2015). Recently proposed Graph Convolution Networks (Kipf and Welling, 2016a) outperforms spatial convolutions and are scalable to large graphs. Various extensions to the GCN model have been proposed for signed networks (Derr, Y. Ma, and Jiliang Tang, 2018), inductive settings (Hamilton, Z. Ying, and Leskovec, 2017) and multiple relations (Schlichtkrull et al., 2018; Zhuang and Q. Ma, 2018) and evolution (Sankar, Yanhong Wu, et al., 2018). All of the GCN variants assume label sharing as they assume similarity between connected nodes.

In Recommender Systems, GCNs have been used to model the user-item interaction graph. GCMC (Berg, Kipf, and Welling, 2017) extends GCN by training an auto-encoder framework on a bipartite user-item interaction graph that performs differentiable message passing, aggregating data from a user's and an item's 'neighbors'. PinSage (R. Ying et al., 2018) proposed a random walk based sampling of neighbors to scale GCNs to web scale graphs. Fan et al. (2019) further extend these methods to incorporate information from a user's social connections. Similarly, Le Wu et al. (2019) use graph neural networks to model diffusion of social influence in recommender systems.

However, these methods either do not take a user's social neighbors into account or operate on static features. All these models also assign uniform weight to all their neighbors, which does not represent online social communities well. Typically in these communities, some friends are only superficially known while others are known personally for years. Thus, they exert a different degree of influence on a user's behavior. Graph Attention Networks (Veličković et al., 2018) can capture the varying influence stengths as they learn attention weights between each pair of nodes in a static graph.

Part II

APPLICATIONS

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3

MODELING RELATIONAL ASPECTS OF USER-GENERATED CONTENT

In this chapter, we propose to incorporate user behavioral information as metadata to improve the attribute estimation of user-generated content. For this purpose, we focus on the task of quality estimation of user-provided answers for best answer selection task in CQA forums. Current approaches predict answer quality in isolation of the other answers to the question and user activity across the forum (other posted questions or answers). This assumption is limiting as the best answer is, in general, selected based on how it differs from other answers to the same question. Similarly, answers given by expert users tend to be of higher quality. We leverage these cues by inducing multiple graphs between these answers based on similarity and contrast in the behavior of users providing the answers. Finally, multiple graphs expressing semantically diverse relationships are merged to predict the best answer to a question (Narang et al., 2019).

3.1 OVERVIEW

Individuals often visit Community Question Answer (CQA) forums, like StackExchange, to seek answers to nuanced questions that are not readily available on web-search engines. Unlike other familiar Learning-to-Rank problems in the IR community (C. J. C. Burges, 2010; C. J. Burges, Ragno, and Le, 2007), CQA platforms can identify and leverage past questions asked by similar users and relevant answers to those questions. However, for CQA sites like StackExchange, individuals who post questions may label answer as 'accepted,' but other questions with answers (about 47% in our analysis) have none labeled as 'accepted.' On other CQA sites like Reddit, there is no mechanism for a person to label an answer as 'accepted.' As a first step to address the individual's information needs, in this work, we focus on the problem of identifying accepted answers on StackExchange.

One approach to identify relevant answers is to identify salient features for each question-answer tuple (q, a) and treat it as a supervised classification problem (Burel, Mulholland, and Alani, 2016; Jenders, Krestel, and Naumann, 2016; Tian and B. Li, 2016; Tian, P. Zhang, and B. Li, 2013).

Deep Text Models further develop this approach (Sukhbaatar et al., 2015; Di Wang and Nyberg, 2015; W. Wu, H. Wang, and X. Sun, 2018; X. Zhang et al., 2017). These models learn the optimal text representation of (q, a) tuple to select the most relevant answer. While the deep text models are sophisticated, text-based models are computationally expensive to train. Furthermore, there are limitations to examining (q, a) tuples in isolation: an answer is "relevant" *in relationship* to other answers to the same question; second, it ignores the fact that same user may answer multiple questions in the forum. These relational aspects of user-generated content provide a unique dimension that is absent in textual search. However, there is only limited work in the context of identification of "best answers" among user-generated content that exploit these implicit and explicit connections. Thus, our key proposal is to use this alternative approach and build a flexible and expressive framework to incorporate the relational aspects of user-generated content for the answer selection task.

Relational aspects are best captured as graphs connecting content. Graph Convolutional Networks (GCNs) is a popular technique to incorporate graph structure, and are used in tasks including node classification (Kipf and Welling, 2016a) and link prediction (Schlichtkrull et al., 2018). Extensions to the basic GCN model include signed networks (Derr, Y. Ma, and Jiliang Tang, 2018), inductive settings (Hamilton, Z. Ying, and Leskovec, 2017) and multiple relations (Schlichtkrull et al., 2018; Zhuang and Q. Ma, 2018). While GCNs are a plausible approach, we need to overcome a fundamental implicit assumption in prior work before we can apply it to our problem. Prior work in GCNs adopt label sharing amongst nodes; label sharing implicitly assumes similarity between two nodes connected by an edge. In the Answer Selection problem, however, answers to the same question connected by an edge may not share the acceptance label. In particular, we may label an answer as 'accepted' based on how it differs from other answers to the same question. In other words, the relational views (or graphs) could capture similarity or contrast between connected content, depending on the relation in consideration. However, Signed GCNs (Derr, Y. Ma, and Jiliang Tang, 2018) can not capture this contrast despite their ability to incorporate signed edges. Graph attention networks (Velickovic et al., 2017) also could not learn negative attention weight over neighbors as weights are the output of a softmax operation.

Thus, we develop a novel framework to model the diverse relations between content through a separate *induced* graph across (q, a) tuples. The key idea is to use diverse strategies—label depends only on the answer (reflexive), the label is determined in contrast with the other answers to the question (contrastive), and label sharing among answers across questions if it contrasts with other answers similarly(similar contrast)—to identify the accepted answer. Each strategy *induces* a graph between (q, a) tuples and then uses a particular label selection mechanism to identify the accepted answer. Our strategies generalize to a broader principle: pick an equivalence relation to induce a graph comprising cliques, and then pick a label selection mechanism (label sharing or label contrast) within each clique. We show how to develop GCN architecture to operationalize the specific label selection mechanism (label sharing or label contrast). Then, we aggregate results across strategies through a boosting framework to identify the label for each (q, a) tuple. Our Contributions are as follows:

MODULAR, INDUCED RELATIONAL FRAMEWORK: We introduce a modular framework that separates the construction of the graph with the label selection mechanism. In contrast, prior work in answer selection (e.g., (Burel, Mulholland, and Alani, 2016; Jenders, Krestel, and Naumann, 2016; Tian and B. Li, 2016; Tian, P. Zhang, and B. Li, 2013).) looked at individual tuples, and work on GCNs (e.g., (Kipf and Welling, 2016a; Zhuang and Q. Ma, 2018)) use the given graph (i.e., no induced graphs) and with similarity as a mechanism for label propagation. We use equivalence relations to induce a graph comprising cliques and identify two label assignment mechanisms—label contrast, label sharing. Then, we show how to encode these assignment mechanisms in GCNs. In particular, we show that the use of equivalence relations allows us to perform *exact* convolution in GCNs. We call our framework Induced Relational GCN (IR-GCN). Our framework allows for parallelization and applies to other problems that need application semantics to induce graphs independent of any existing graphs (Brugere, Gallagher, and Berger-Wolf, 2018).

DISCRIMINATIVE SEMANTICS: We show how to encode the notion of label contrast between a vertex and a group of vertices in GCNs. Label contrast is critical to the problem of best answer selection. Related work in GCNs (e.g., (Kipf and Welling, 2016a; Zhuang and Q. Ma, 2018)) emphasizes node similarity, including the work on signed graphs (Derr, Y. Ma, and Jiliang Tang, 2018). In (Derr, Y. Ma, and Jiliang Tang, 2018), contrast is a property of an edge, not a group, and is not expressive enough for our problem. We show that our encoding of contrast creates *discriminative magnification*—the separation between nodes in the embedding space is most meaningful at smaller clique sizes; the effect decreases with clique size.

BOOSTED ARCHITECTURE: We show through extensive empirical results that using common boosting techniques improves learning in our convolutional model. This improvement is a surprising result since much of the work on neural architectures develops stacking, fusion, or aggregator architectures.

We conducted extensive experiments using our IR-GCN framework with excellent experimental results on the popular CQA forum—StackExchange. For our analysis, we collect data from 50 communities—the ten largest communities from each of the five StackExchange (https://stackexchange.com/sites) categories. We achieved an improvement of over 4% accuracy and 2.5% in MRR, on average, over state-of-the-art baselines. We also provide Reddit (https://www.reddit.com/) results using expert answers as a proxy for acceptance, to overcome the absence of explicit labels. Finally, we show that our model is more robust to label sparsity compared to alternate GCN based multi-relational approaches.

We organize the rest of this chapter as follows. In section 3.2, we formulate our problem statement and then discuss induced relations for the Answer Selection problem in section 3.3. We then detail the operationalization of these induced relations in the Graph Convolution framework in section 3.4 and introduce our gradient boosting based aggregator approach in section 3.5. Section 3.6 describes experiments and Section 3.7 describes further ablation studies. We finally conclude in section 3.8.

3.2 PROBLEM FORMULATION

In Community Question Answer (CQA) forums, an individual asking a question seeks to identify the most relevant candidate answer to his question. On Stack-Exchange CQA forums, users annotate their preferred answer as “accepted.”

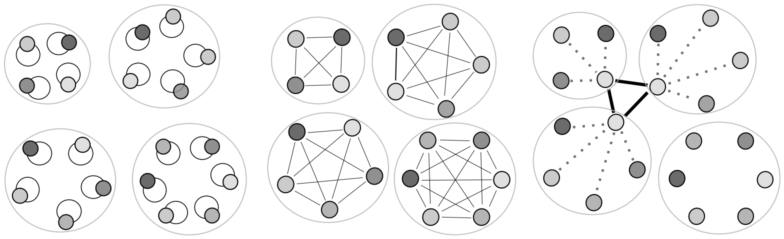
Let $\mathcal{Q}$ denote the set of questions in the community and for each $q \in \mathcal{Q}$, we denote $\mathcal{A}_q$ to be the associated set of answers. Each question $q \in \mathcal{Q}$, and each answer $a \in \mathcal{A}_q$ has an author $u_q, u_a \in \mathcal{U}$ respectively. Without loss of generality, assume that we can extract features for each question q, each answer $a \in \mathcal{A}_q$, user $u_q, u_a \in \mathcal{U}$.

Our unit of analysis is a question-answer tuple $(q, a), q \in \mathcal{Q}, a \in \mathcal{A}_q$, and we associate each (q, a) tuple with a label $y_{q,a} \in \{-1, +1\}$, where ‘+1’ implies acceptance and ‘-1’ implies rejection.

> The goal of this work is to develop a framework to identify the accepted answer to a question posted on a CQA forum.

3.3 INDUCED RELATIONAL VIEWS

In this section, we discuss the idea of induced relational views, central to our induced relational GCN framework developed in Section 3.4. First, in Section 3.3.1, we introduce potential strategies for selecting the accepted answer given a question. We show how each strategy induces a graph G on the question-answer (q, a) tuples. Next, in Section 3.3.2, we show how each of these example strategies is an instance of an equivalence relation; our framework generalizes to incorporate any such relation.



(a) Reflexive (b) Contrastive (c) Similar Contrast

Figure 3.1: Reflexive(fig. 3.1a), Contrastive (fig. 3.1b) and Similar Contrast (fig. 3.1c) relations among (q, a) tuples. Reflexive assumes no dependence on other answers for prediction. Contrastive compares between all answers to a question; Similar Contrast connects answers across questions if they contrasts with other answers similarly. Solid lines show the similarity relation while dotted lines signify the contrast. The contrast is only significant in three questions.

3.3.1 *Constructing Induced Views*

In this section, we discuss in detail four example strategies that can be used by the individual posting the question to label an answer as 'accepted.' Each of the $S_i \in \mathbf{S}$ strategies *induces* a graph $G_i = (V, E_i)$ (also referred to as a relational view). In each graph G_i, a vertex $v \in V$ corresponds to a tuple (q, a) and an edge $e \in E_i, E_i \subseteq V \times V$ connects two tuples that are matched under that strategy. Note that each G_i has the same vertex set V, and the edge sets E_i are strategy dependent. Each strategy employs one of the three different relation types—reflexive, contrastive, and similar—to connect the tuples. We use one reflexive strategy, one contrastive, and two similar strategies. Figure 3.1 summarizes the three relations. Below, we organize the discussion by relation type.

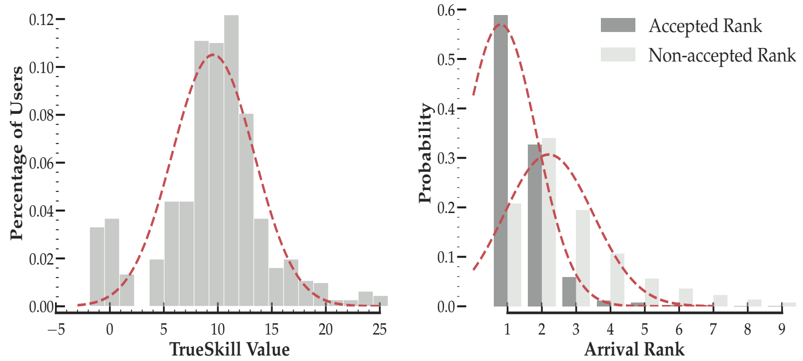
REFLEXIVE: A natural strategy is to examine each (q, a) tuple in isolation and then assign a label $y_{q,a} \in \{-1, +1\}$ corresponding to 'not accepted' or 'accepted.' In this case, $y_{q,a}$ depends on only the features of (q, a). This is a *Reflexive* relation, and the corresponding graph $G_r = (V, E_r)$ has a specific structure. In particular, in this graph G_r, we have only self-loops, and all edges $e \in E_r$ are of the type (v, v). That is, for each vertex $v \in V$, there are no edges (v, u) to any other vertices $u \neq v \in V$. Much of the prior work on feature driven answer selection (Burel, Mulholland, and Alani, 2016; Jenders, Krestel, and Naumann, 2016; Tian and B. Li, 2016; Tian, P. Zhang, and B. Li, 2013) adopts this view.

CONTRASTIVE: A second strategy is to examine answers *in relation* to other answers to the same question and label one such answer as 'accepted.' Thus the second strategy *contrasts* (q, a), with other tuples in $(q, a'), q \in \mathcal{Q}; a, a' \in \mathcal{A}_q; a' \neq a$. This is a *Contrastive* relation and the corresponding graph $G_c = (V, E_c)$ has a specific structure. Specifically, we define an edge $e \in E_c$ for all (q, a) tuples for the same question $q \in \mathcal{Q}$. That is, if $v = (q_1, a_1), u = (q_2, a_2), e = (u, v) \in E_c \iff q_1 = q_2$. Intuitively, the contrastive relation induces cliques connecting all answers to the same question. Introducing contrasts between vertices sharpens differences between features, an effect (described in more detail in Section 3.4.2) we term *Discriminative Feature Magnification*. Notice that the contrastive relation is distinct from graphs with signed edges (e.g., (Derr, Y. Ma, and Jiliang Tang, 2018)). In our framework, the contrast is a *neighborhood* property of a vertex, whereas in (Derr, Y. Ma, and Jiliang Tang, 2018), the negative sign is a property of an *edge*.

SIMILAR CONTRASTS: A third strategy is to identify *similar* (q, a) tuples *across* questions. Prior work (Lingfei Wu, Baggio, and Janssen, 2016) indicates that individuals on StackExchange use diverse strategies to contribute answers. Experts (with a high reputation) tend to answer harder questions, while new members (with low reputation) looking to acquire reputation tend to be the first to answer a question.

How might similarity by contrast work? Consider two individuals Alice and Bob with *similar* reputations (either high or low) on StackExchange, who contribute answers a_A and a_B to questions q_1 and q_2 respectively. If Alice and Bob have high reputation difference with other individuals who answer questions q_1 and q_2 respectively, then it is likely that (q_1, a_A) and (q_2, a_B) will share the same label (if they are both experts, their answers might be accepted, if they are both novices, then this is less likely). However, if Alice has a high reputation difference with other peers who answer q_1, *but Bob does not have that difference* with peers who answer q_2, then it is less likely that the tuples (q_1, a_A) and (q_2, a_B) will share the label, even though the reputations of Alice and Bob are similar.

Thus, the key idea of the *Similar Contrasts* relation is that link tuples that are *similar in how they differ* with other tuples. We construct the graph $G_s = (V, E_s)$ in the following manner. An edge $e = (v, u)$ between tuples v and u exists if the similarity $s(v, u)$ between tuples v, u exceeds a threshold δ. We define the similarity function $s(\cdot, \cdot)$ to encode similarity by contrast. That is, $e = (v, u) \in E_s \iff s(v, u) \geq \delta$.



(a) TrueSkill Distribution

(b) ArrivalRank Distribution

Figure 3.2: Distribution of the TrueSkill values of users and ArrivalRank of accepted answers and non-accepted answers for the movie StackExchange. Early answers are more likely to be accepted and variance of TrueSkill similarity across users is high.

Motivated by (Lingfei Wu, Baggio, and Janssen, 2016) and our empirical analysis (fig. 3.2), we consider two different views that correspond to the similar contrast relation. The *TrueSkill Similarity* view connects all answers authored by a user where her skill (computed via Bayesian TrueSkill (Herbrich, Minka, and Graepel, 2006))) differs from competitors by margin δ. We capture both cases when the user is less or more skilled than her competitors. Under this view, we connect answers authored by a specific user, where the difference in his skill over peers is greater than margin δ. Specifically, if the user authors answers a, a' to questions q, q', we create a link between a and a' if

$$|S_{u,a} - S_{u,b}| > \delta; \forall b \in \mathcal{A}_q \tag{3.1}$$

$$|S_{u,a'} - S_{u,c}| > \delta; \forall c \in \mathcal{A}_{q'} \tag{3.2}$$

where $S_{u,a}$ is the skill value for the user who authored answer a. Similarly, a link is created for the opposite case when difference is less than

$-\delta$. We estimate the user skill values with the TrueSkill rating system (`https://pypi.org/project/trueskill/`) computed from their historic performance in the community. TrueSkill values are normally distributed among users (fig. 3.2a).

In the *Arrival Similarity* view, we connect answers across questions based on the similarity in the relative time of their arrival (posting timestamp). The temporal arrival patterns of answers are correlated to their acceptance probabilities (fig. 3.2b). For a specific user authoring answers a, a' to questions q, q', we establish a link between these answers if

$$|T_a - T_b| > \gamma \times \max(T_b); \forall b \in \mathcal{A}_q \tag{3.3}$$

$$|T_{a'} - T_c| > \gamma \times \max(T_c); \forall c \in \mathcal{A}_{q'} \tag{3.4}$$

where T_a represents the relative time-gap between answer a and the question q. Conversely, we create links when difference is less than $-\gamma \times \max(T_b)$.

We hypothesize that a similar answering schedule indicates similar user confidence or skill across questions. Notice that two Similar Contrast views have different edge (E) sets since the corresponding similarity functions are different. Notice also, that the two similarity function definitions are transitive. [1]

3.3.2 *Generalized Views*

Now we present the general case of the induced view. First, notice that each of the three relation types that we consider—reflexive, contrastive, and similar—result in a graph $G_i = (V, E_i)$ comprising a set of cliques. The resulting set of cliques is not surprising, since all three relations presented here, are equivalence relations. Second, observe the semantics of how we select the tuple with the accepted answer. Within the three relations, we used two semantically different ways to assign the 'accepted' answer label to a tuple. One way is to share the labels amongst all the vertices in the *same clique* (used in the reflexive and the similar relations). The second is to *assign label based on contrasts with other vertices* in the same clique. We can now state the organizing principle of our approach as follows.

[1] One trivial way of establishing similarity is co-authorship i.e., connect all (q, a) tuples of a user (probably on the same topic) across different questions. Note that the accepted answer is labeled relative to the other answers. As the competing answers are different in each question, we can not trivially assume acceptance label similarity for all coauthored answers. In our experiments, co-authorship introduced a lot of noisy links in the graph leading to worse performance.

> A generalized *modular* framework: pick a meaningful equivalence relation on the (q, a) tuples to induce graph comprising cliques and then apply specific label semantics within each clique.

Equivalence relation results in a graph with a set of disconnected cliques. Then, within a clique, one could use application-specific semantics, different from two discussed in this paper, to label tuples as 'accepted.' Cliques have some advantages: they have well-defined graph spectra Chung, 1997, p. 6; cliques allows for *exact* graph convolution; parallelize the training as the convolution of a clique is independent of other cliques.

Thus, each strategy induces a graph $G_i = (V, E_i)$ using one of the three equivalence relations—reflexive, contrastive, and similar—and then applies one of the two semantics ('share the same label'; 'determine label based on contrast').

3.4 INDUCED RELATIONAL GCN

Now, we will encode the two label assignment mechanisms within a clique via a graph convolution. First, we briefly review Graph Convolution Networks (GCN) and identify some key concepts. Then, given the views G_i for the four strategies, we show how to introduce label contrasts in Section 3.4.2 followed by label sharing in Section 3.4.3.

3.4.1 *Graph Convolution*

Graph Convolution models adapt the convolution operations on regular grids (like images) to irregular graph-structured data $G = (V, E)$, learning low-dimensional vertex representations. If for example, we associate a scalar with each vertex $v \in V$, where $|V| = N$, then we can describe the convolution operation on a graph by the product of signal $x \in \mathbb{R}^N$ (feature vectors) with a learned filter g_θ in the fourier domain. Thus,

$$g_\theta * x = U\, g_\theta\, U^T x, \tag{3.5}$$

where, Λ and U are the eigenvalues and eigenvector of the normalized graph Laplacian, $L = I_N - D^{-1/2} A D^{1/2}$, and where $L = U \Lambda U^T$. A denotes the adjacency matrix of a graph G (associated with a view) with N vertices. Equation (3.5) implies a filter g_θ with N free parameters, and requires expensive eigenvector decomposition of the adjacency matrix A. Defferrard, Bresson, and Vandergheynst (2016) proposed to approximate g_θ, which in general is a function of Λ, by a sum of Chebyshev polynomials $T_k(x)$ up to the k-th order. Then,

$$g_\theta * x \approx U \sum_{k=0}^{K} \theta_k T_k(\tilde{\Lambda})\, U^T x \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L})\, x, \tag{3.6}$$

where, $\tilde{\Lambda} = 2\Lambda/\lambda_{\max} - I_N$ are the scaled eigenvalues and $\tilde{L} = 2L/\lambda_{max} - I_N$ is the corresponding scaled Laplacian. Since $\tilde{L} = U\tilde{\Lambda}U^T$, the two equations are approximately equal.

The key result from Defferrard, Bresson, and Vandergheynst (2016) is that Equation (3.6) implies k-hop localization—the convolution result depends only on the k-hop neighborhood. In other words, Equation (3.6) is a k-hop approximation.

However, since we use equivalence relations in our framework that result in cliques, we can do an *exact* convolution operation since vertices in a clique only have one-hop (i.e., $k = 1$) neighbors (see lemma 5.2, (Hammond, Vandergheynst, and Gribonval, 2011)). The resulting convolution is linear in L and now has only two filter parameters, θ_0 and θ_1 shared over the whole graph.

$$g_\theta * x = \theta_0 x + \theta_1 (L - I_N)\, x \tag{3.7}$$

Our formulation is exact since we perform the convolution on cliques.

We emphasize the distinction with Kipf and Welling (2016a) who approximate the Defferrard, Bresson, and Vandergheynst (2016) observation by restricting $k = 1$. They do so since they work on arbitrary graphs; since our relations result in views with cliques, we do not make any approximation by using $k = 1$.

3.4.2 *Contrastive Graph Convolution*

Now, we show how to perform graph convolution to encode the mechanism of contrast, where label assignments for a tuple depend on the contrast with its neighborhood.

To establish contrast, we need to compute the *difference* between the vertex's own features to its neighborhood in the clique. Thus we transform Equation (3.7) by setting $\theta = \theta_0 = \theta_1$, which essentially restricts the

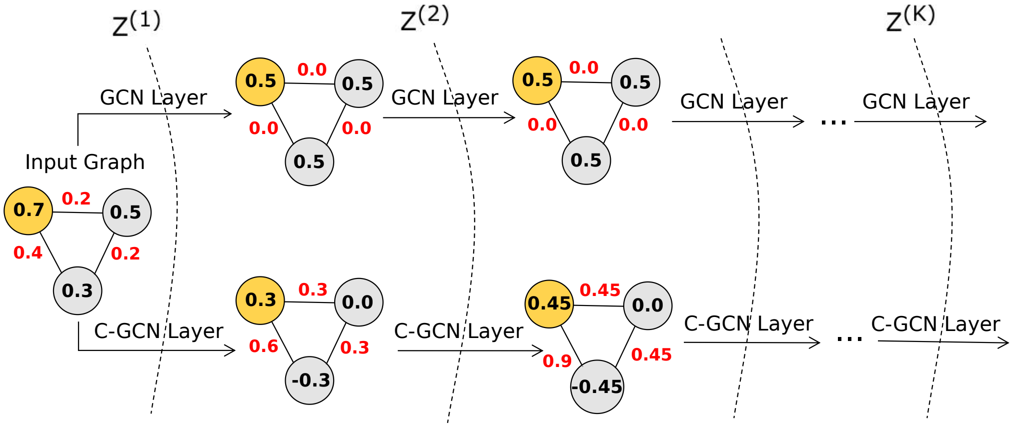


Figure 3.3: Stylized example showing the convolution results of GCN and proposed Contrastive GCN for a question with three answers. Edge labels denote the feature difference while node labels denote the resulting feature value. The feature difference between neighboring nodes increases with each convolution layer for Contrastive GCN while GCN averages the feature values among nodes.

filters learned by the GCN. This transformation leads to the following convolution operation:

$$g_\theta * x = \theta\left(I_N + L - I_N\right) x \tag{3.8}$$

$$g_\theta * x = \theta\left(I_N - D^{-1/2} A D^{-1/2}\right) x \tag{3.9}$$

Notice that Equation (3.9) says that for example, for any vertex u with a scalar feature value x_u, for a given clique with $n \geq 2$ vertices, the convolution operation computes a new value $\hat{x}_u$ for vertex u as follows:

$$\hat{x}_u = \theta\left(x_u - \frac{1}{n-1} \sum_{v \in \mathcal{N}_u} x_v\right). \tag{3.10}$$

where $\mathcal{N}_u$ is the neighborhood of vertex u. Notice that since our equivalence relations construct cliques, for all vertices u that belong to a clique of size n, $|\mathcal{N}_u| = n - 1$.

When we apply the convolution operation in Equation (3.9) at each layer of GCN, output for the k-th layer is:

$$\mathbf{Z}_c^k = \sigma\left(\left(I_N - D^{-1/2} A_c D^{1/2}\right) \mathbf{Z}_c^{k-1} \mathbf{W}_c^k\right) \tag{3.11}$$

with A_c denoting the adjacency matrix in the contrastive view. $\mathbf{Z}_c^k \in \mathbb{R}^{N \times d}$ are the learned vertex representations for each (q, a) tuple under the contrastive label assignment. N is the total number of tuples and d refers to the dimensionality of the embedding space. $\mathbf{Z}^{k-1}$ refers to the output of the previous $(k-1)$-th layer, and $\mathbf{Z}^0 = X$ where X is the input feature matrix. $\mathbf{W}_c^k$ are the filter θ parameters learnt by the GCN; $\sigma(\cdot)$ denotes the activation function (e.g. ReLU, tanh).

To understand the effect of Equation (3.11) on a tuple, let us restrict our attention to a vertex u in a clique of size n. We can do this since the convolution result in one clique is unaffected by other cliques. When we do this, we obtain:

$$z_c^k(u) = \sigma\left(\left(z_c^{k-1}(u) - \frac{1}{n-1} \sum_{v \in \mathcal{N}_u} z_c^{k-1}(v)\right) \mathbf{W}_c^k\right). \tag{3.12}$$

Now consider a pair of contrasting vertices, u and v in the same clique of size n. Let us ignore the linear transform by setting $W_c^k = \mathbf{I}$ and set $\sigma(\cdot)$ to the identity function. Then we can easily verify that:

$$z_c^k(u) - z_c^k(v) = \underbrace{\left(1 + \frac{1}{n-1}\right)}_{\text{magnification}} \times \underbrace{\left(z_c^{k-1}(u) - z_c^{k-1}(v)\right)}_{\text{contrast in previous layer}}, \tag{3.13}$$

where, $z_c^k(u)$ denotes the output of the k-th convolution layer for the u-th vertex in the contrastive view. As a result, each convolutional layer magnifies the feature contrast between the vertices that belong to the same clique. Thus, the contrasting vertices move further apart. We term this as *Discriminative Feature Magnification* and Equation (3.13) implies that we should see higher magnification effect for smaller cliques. An illustration is provided in the bottom part of the fig. 3.3 with a uni-dimensional feature.

Contrasting nodes are shifted further apart by eq. (3.11) improving their separability in the learned manifold (further discussion in section 3.7.6).

3.4.3 *Encoding Similarity Convolution*

We next discuss how to encode the mechanism of sharing labels in a GCN. While label sharing applies to our similar contrast relation (two strategies: Arrival similarity; TrueSkill similarity, see Section 3.3.1), it is also trivially applicable to the reflexive relation, where the label of the tuple only depends on itself. First, we discuss the case of similar contrasts.

encoding similar contrasts: To encode label sharing for the two similar by contrast cases, we transform Equation (3.7) with the assumption $\theta = \theta_0 = -\theta_1$. Thus

$$g_\theta * x = \theta \left(I_N + D^{-1/2} A D^{-1/2} \right) x, \tag{3.14}$$

Similar to the Equation (3.9) analysis, convolution operation in Equation (3.14) computes a new value $\hat{x}_u$ for vertex u as follows:

$$\hat{x}_u = \theta \left(x_u + \frac{1}{n-1} \sum_{v \in \mathcal{N}_u} x_v \right). \tag{3.15}$$

$$\hat{x}_u = \theta \left(\frac{n-2}{n-1} x_u + \frac{n}{n-1} \mu_x \right). \tag{3.16}$$

That is, in the mechanism where we share labels in a clique, the convolution pushes the values of each vertex in the clique to the average feature value, $\mu_x = \frac{1}{n} \sum_{v \in \mathcal{N}_u \cup u} x_v$, in the clique.

When we apply the convolution operation in Equation (3.14) at each layer of GCN, output for the k-th layer:

$$\mathbf{Z}_s^k = \sigma \left(\left(I_N + D^{-1/2} A_s D^{1/2} \right) \mathbf{Z}_s^{k-1} \mathbf{W}_s^k \right) \tag{3.17}$$

with A_s denoting the adjacency matrix in the similar views.

We analyze the similarity GCN in a maner akin to Equation (3.12) and we can easily verify that:

$$z_s^k(u) - z_s^k(v) = \underbrace{\left(1 - \frac{1}{n-1} \right)}_{\text{reduction}} \times \underbrace{\left(z_s^{k-1}(u) - z_s^{k-1}(v) \right)}_{\text{contrast in previous layer}}, \tag{3.18}$$

where, $z_s^k(i)$ denotes the output of the k-th convolution layer for the i-th vertex in the similar view. As a result, each convolutional layer reduces the feature contrast between the vertices that belong to the same clique. Thus, the similar vertices move closer (see top part in fig. 3.3).

The proposed label sharing encoding applies to both similar contrast strategies (TrueSkill; Arrival). We refer to the corresponding vertex representations as $\mathbf{Z}_{ts}^k$ (TrueSkill), $\mathbf{Z}_{as}^k$ (Arrival).

REFLEXIVE CONVOLUTION: We encode the reflexive relation with self-loops in the graph resulting in an identity adjacency matrix. This relation is the trivial label sharing case, with an independent assignment of vertex labels. Thus, the output of the k-th convolutional layer for the reflexive view, $\mathbf{Z}_r^k$ reduces to:

$$\mathbf{Z}_r^k = \sigma\left(I_N \mathbf{Z}_r^{k-1} \mathbf{W}_r^k\right) \tag{3.19}$$

Hence, the reflexive convolution operation is equivalent to a feedforward neural network with multiple layers and activation $\sigma(\cdot)$.

Each strategy $S_i \in \mathbf{S}$ belongs to one of the three relation types—reflexive, contrastive and similarity, where $\mathbf{R}$ denotes the set of strategies of that relation type. $\mathcal{R} = \bigcup \mathbf{R}$ denotes the set of all relation types. $\mathbf{Z}_i^K \in \mathbb{R}^{NXd}$ represents the d dimensional vertex embeddings for strategy S_i at the K-th layer. For each strategy S_i, we obtain a scalar score by multiplying $\mathbf{Z}_i^K$ with transform parameters $\widetilde{W}_i \in \mathbb{R}^{d\times 1}$. The sum of these scores gives the combined prediction score, $\mathbf{H}_\mathbf{R} \in \mathbb{R}^{NX1}$, for that relation type.

$$\mathbf{H}_\mathbf{R} = \sum_{S_i \in \mathbf{R}} \mathbf{Z}_i^K \widetilde{W}_i^T \tag{3.20}$$

In this section, we proposed Graph Convolutional architectures to compute vertex representations of each (q, a) tuple under the four strategies. In particular, we showed how to encode two different label assignment mechanisms—label sharing and determine label based on contrast—within a clique. The architecture that encodes label assignment based on contrast is a novel contribution; distinct from the formulations presented by Kipf and Welling (2016a) and its extensions (Derr, Y. Ma, and Jiliang Tang, 2018; Schlichtkrull et al., 2018). Prior convolutional architectures implicitly encode the label sharing mechanism (eq. (3.14)); however, label sharing is unsuitable for contrastive relationships across vertices. Hence our architecture fills this gap in prior work.

3.5 AGGREGATING INDUCED VIEWS

In the previous sections, we introduced four strategies to identify the accepted answer to a question. Each strategy induces a graph or relational view between (q, a) tuples. Each relational view is expected to capture semantically diverse neighborhoods of vertices. The convolution operator aggregates the neighborhood information under each view. The key question that follows is, *how do we combine these diverse views in a unified learning framework?* Past work has considered multiple solutions:

NEIGHBORHOOD AGGREGATION : In this approach, they represent vertices by aggregating feature representations of it's neighbors across all views (Hamilton, Z. Ying, and Leskovec, 2017; Schlichtkrull et al., 2018). Specifically, the final adjacency matrix is the sum of all the individual adjacency matrices of each view, i.e., $A = \sum_{S_i \in \mathbf{S}} A_i$. They, then, apply Graph Convolution Network to this updated Adjacency matrix.

STACKING : Multiple convolution layers stacked end-to-end (each potentially handling a different view) (Yu, Yin, and Zhu, 2017). Specifically, they stacks all GCNs belonging to a view such that output of a lower GCN is fed as an input to the GCN directly above it. Thus, output from the last layer of GCN for view i, Z_i^K s.t. $S_i \in \mathbf{S}$ will act as input features, Z_j^0 for some other view j s.t. $S_j \in \{\mathbf{S} - S_i\}$ if view j is directly above the view i. In our experiments, we obtain the best performance by using the following order: Contrastive, Similarity by Contrast followed by Reflexive.

FUSION : Follows a multi-modal fusion approach (Farnadi et al., 2018), where views are considered distinct data modalities. It treats each GCN as a separate model and appends the output from the final layer of each GCN i.e. $Z_i^K; \forall S_i \in \mathbf{S}$ to the input of all the other GCN's, i.e. $Z_j^0 \forall S_j \in \mathbf{S} - S_i$ along with the original features. Thus, the input of each GCN is linear in $|\mathbf{S}|$.

SHARED LATENT STRUCTURE : Attempts to transfer knowledge across relational views (modalities) with constraints on the representations (e.g. (Zhuang and Q. Ma, 2018) aligns embeddings across views).

Ensemble methods introduced in (Schlichtkrull et al., 2018) work on multi-relational edges in knowledge graphs. None of these approaches are directly suitable for our induced relationships. Our relational views utilize different label assignment semantics (label sharing within a clique vs. determine label based on contrast within a clique). In our label contrast semantics, we must achieve feature discrimination and label inversion between contrasting vertices, as opposed to label homogeneity and feature sharing in the label sharing case. Thus, aggregating relationships by pooling, concatenation, or addition of vertex representations fail to capture semantic heterogeneity of the induced views. Further, data induced relations are uncurated and inherently noisy. Directly aggregating the learned representations via Stacking or Fusion can lead to noise propagation. We also expect views of the same relation type to be correlated.

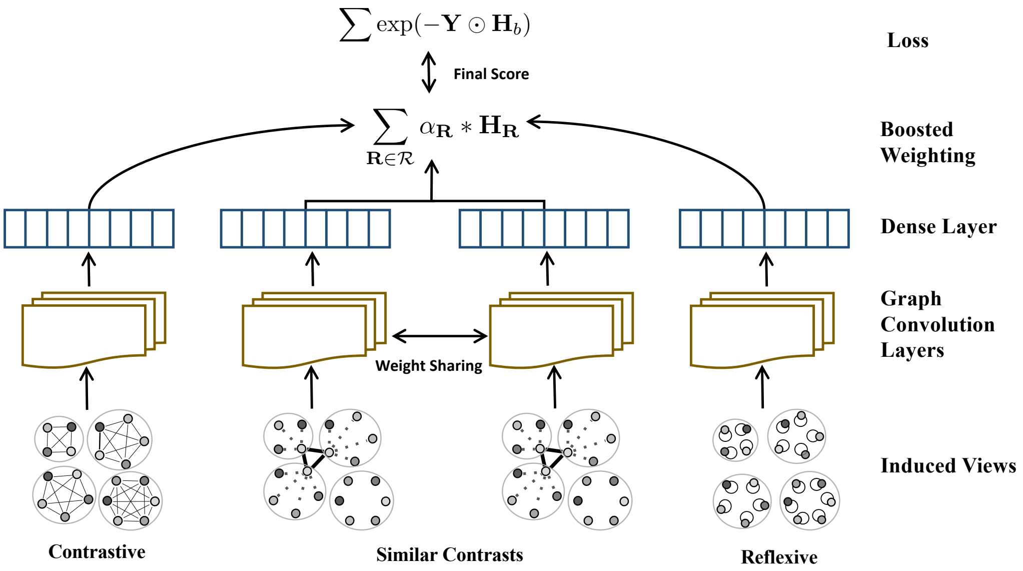


Figure 3.4: Schematic diagram of our proposed IR-GCN model.

We thus propose the following approach to aggregate information across relation types and between views of a relation type.

CROSS-RELATION AGGREGATION: We expect distinct relation types to perform well on different subsets of the set of (q, a) tuples. We empirically verify this with the Jaccard overlap between the set of misclassified vertices under each relational view of a relation type on our dataset. Given $\mathbf{M}_A$ and $\mathbf{M}_B$, the sets of (q, a) tuples misclassified by GCNs A and B respectively, the jaccard overlap is,

$$\mathcal{J}_{A,B} = \frac{\mathbf{M}_A \cap \mathbf{M}_B}{\mathbf{M}_A \cup \mathbf{M}_B} \tag{3.21}$$

The $\mathcal{J}_{A,B}$ values are as follows for the relational pairings: (Contrastive, TrueSkill Similarity) = 0.42, (Contrastive, Reflexive) = 0.44 and (Reflexive, TrueSkill Similarity) = 0.48. Relatively low values of the overlap metric indicate uncorrelated errors across the relations.

Gradient boosting techniques are known to improve performance when individual classifiers, including neural networks (Schwenk and Bengio, 2000), are diverse yet accurate. A natural solution then is to apply boosting to the set of relation types and bridge the weaknesses of each learner. We employ Adaboost (Freund and Schapire, 1995) to combine relation level scores, $\mathbf{H_R}$ (eq. (3.20)) in a weighted manner to compute the final boosted score, $\mathbf{H}_b \in \mathbb{R}^{N\times 1}$ representing all relation types (Line 12, algorithm 3.1). $\mathbf{Y} \in \mathbb{R}^{NX1}$ denotes the acceptance label of all tuples. Note that an entry in $(\mathbf{Y} \odot \mathbf{H_R}) > 0$ when the accepted label of the corresponding (q, a) tuple and sign of the prediction score, $sign(\mathbf{H_R})$, of relation type $\mathbf{R}$ match and < 0 otherwise. Thus, the weights $\alpha_\mathbf{R}$ adapt to the fraction of correctly classified tuples to the misclassified tuples by the relation $\mathbf{R}$ (Line 9, algorithm 3.1). The precise score computation is described in algorithm 3.1. We use the polarity of each entry in the boosted score, $sign(\mathbf{H}_b) \in \{-1, 1\}$, to predict the class label of the corresponding (q, a) tuple. The final score is also used to create a ranked list among all the candidate answers, $a \in \mathcal{A}(q)$ for each question, $q \in \mathcal{Q}$. $L_{(q,a)}$ represents the position of candidate answer a in the ranked list for question q.

INTRA-RELATION AGGREGATION : Gradient boosting methods can effectively aggregate relation level representations, but are not optimal within a relationship type (since it cannot capture shared commonalities between different views of a relation type). For instance, we should facilitate information sharing between the TrueSkill similarity and Arrival similarity views. Thus, if an answer is authored by a user with a higher skill rating and answered significantly earlier than other answers, its probability to be accepted should be mutually enhanced by both signals. Empirically, we also found True Skill and Arrival Similarity GCNs to commit similar mistakes ($\mathcal{J}_{TS,AS} = 0.66$). Thus, intra-relation learning (within a single relation type like Similar Contrast) can benefit from sharing the structure of their latent spaces i.e., weight parameters of GCN.

WEIGHT SHARING : For multiple views representing a relation type (e.g., TrueSkill and Arrival Similarity), we train a separate GCN for each view but share the layer-wise linear-transforms $\mathbf{W}_i^k$ to capture similarities in the learned latent spaces. Weight sharing is motivated by a similar idea explored to capture local and global views in (Zhuang and Q. Ma, 2018). Although sharing the same weight parameters, each GCN can still learn distinct vertex representations as each view convolves over a different neighborhood and employ random dropout during training. We thus propose to use an alignment loss term to minimize prediction difference between views of a single relation type(Sajjadi, Javanmardi, and Tasdizen, 2016). The loss attempts to align the learned vertex representations at the *last layer K* (the loss term aligns pairs of final vertex representations, $||\mathbf{Z}_i^K - \mathbf{Z}_{i'}^K|| \ \forall \ S_i, S_i' \in \mathbf{R}$). In principle, multiple GCNs augment performance of

Algorithm 3.1 IR-GCN Boosted Score Computation

```
function Forward($\mathbf{X}, \mathbf{Y}, \{A_i\}_{S_i \in \mathbf{S}}$)
    $\mathbf{H}_b \leftarrow \mathbf{0}$
    for $\mathbf{R} \in \mathcal{R}$ do
        $\{\mathbf{Z}_i^K\}_{S_i \in \mathbf{R}} \leftarrow Conv(\mathbf{X}, \{A_i\}_{S_i \in \mathbf{R}})$
                                                        ▷ Equation 3.11, 3.17, 3.19
        $\mathbf{H_R} = \sum_{S_i \in \mathbf{R}} \mathbf{Z}_i^K \times \widetilde{\mathbf{W}}_i$                        ▷ Equation 3.20
        $\mathbf{e_R} \leftarrow \exp(-\mathbf{Y} \odot \mathbf{H}_b)$
                                                        ▷ $\odot \longrightarrow$ *Hadamard Product*
        $\alpha_{\mathbf{R}} \leftarrow \frac{1}{2} \ln \frac{\sum \mathbf{e_R} \odot \mathbb{1}((\mathbf{Y} \odot \mathbf{H_R}) > 0)}{\sum \mathbf{e_R} \odot \mathbb{1}((\mathbf{Y} \odot \mathbf{H_R}) < 0)}$
                                                        ▷ $\sum \longrightarrow$ *reduce-sum*
                                        ▷ $\mathbb{1}(.) \longrightarrow$ *element-wise Indicator function*
        $\mathbf{H}_b \leftarrow \mathbf{H}_b + \alpha_{\mathbf{R}} * \mathbf{H_R}$                  ▷ Update boosted GCN
    end for
    return $\mathbf{H}_b, \{\mathbf{H}_R\}_{\mathbf{R} \in \mathcal{R}}, \{\mathbf{Z}_i^K\}_{S_i \in \mathbf{S}}$
                                        ▷ Boosted scores, Relation level scores,
                                            ▷ Each GCN vertex representations
end function
```

the relation type by sharing prior knowledge through multiple Adjacency matrices ($\mathbf{A}_i \ \forall \ S_i \in \mathbf{R}$).

TRAINING ALGORITHM: Algorithm 3.2 describes the training algorithm for our IR-GCN model. For each epoch, we first compute the aggregated prediction score $\mathbf{H}_b$ of our boosted model as described in algorithm 3.1. We use a supervised exponential loss $\mathcal{L}_b$ for training with elastic-net regularization (L1 loss - $\mathcal{L}_1(.)$ and L2 loss - $\mathcal{L}_2(.)$) on the graph convolutional weight matrices $\mathbf{W_i^k} \ \forall \ S_i \in \mathbf{S}$ for each view. Note that we employ weight sharing between all views of the same relation type so that only one set of weight matrices is learned per relation.

The exponential loss, $\mathcal{L}_{\mathbf{R}}$, for each relation type is added alternatingly to the boosted loss. We apply an *exponential annealing schedule*, $\lambda(t)$, i.e. a function of the training epochs (t), to the loss function of each relation. As training progresses and the boosted model learns to optimally distribute vertices among the relations, increase in $\lambda(t)$ ensures more emphasis is provided to the individual convolutional networks of each relation. Figure 3.4 illustrates the overall architecture of our IR-GCN model.

Algorithm 3.2 IR-GCN Training

```
Input: Input Feature Matrix $X$, Acceptance labels for each tuple, $\mathbf{Y}$, Adjacency matrix of each view $\{A_i\}_{S_i \in \mathbf{S}}$
Output: Trained Model i.e. Weight parameters $W_i^1 \dots W_i^k, S_i \in \mathbf{S}, \forall k \in [1, K]$ and transform parameters $\widetilde{W}_i$, $S_i \in \mathbf{S}$
for $t \leftarrow 1$ to num-epochs do
    $\mathbf{H}_b, \{\mathbf{H}_R\}_{\mathbf{R} \in \mathcal{R}}, \{\mathbf{Z}_i^K\}_{S_i \in \mathbf{S}} \leftarrow$ Forward$(X, Y, \{A_i\}_{S_i \in \mathbf{S}})$
                                                            ▷ Algorithm 3.1
    for $\mathbf{R} \in \mathcal{R}$ do
        $\mathcal{L}_b \leftarrow \sum \exp(-\mathbf{Y} \odot \mathbf{H}_b) + \gamma_1 \mathcal{L}_1(.) + \gamma_2 \mathcal{L}_2(.)$
                                                            ▷ $\sum \longrightarrow$ reduce-sum
                                                            ▷ $\odot \longrightarrow$ Hadamard Product
        $\mathcal{L}_\mathbf{R} \leftarrow 0$
        for $S_i \in \mathbf{R}$ do
            $\mathcal{L}_i \leftarrow \sum \exp(-\mathbf{Y} \odot \mathbf{H}_\mathbf{R})$
            $\mathcal{L}_\mathbf{R} \leftarrow \mathcal{L}_\mathbf{R} + \mathcal{L}_i + \frac{1}{2} \sum_{S_i' \neq S_i} ||\mathbf{Z}_i^K - \mathbf{Z}_{i'}^K||$
        end for
        $\mathcal{L}_b \leftarrow \mathcal{L}_b + \lambda(t)\mathcal{L}_\mathbf{R}$
        $W_i^k \leftarrow W_i^k + \eta_{\text{ADAM}} \frac{\partial \mathcal{L}_b}{\partial W_i^k}$          ▷ $\forall k \in [1, K], \forall S_i \in \mathbf{R}$
        $\widetilde{W}_i \leftarrow \widetilde{W}_i + \eta_{\text{ADAM}} \frac{\partial \mathcal{L}_b}{\partial \widetilde{W}_i}$          ▷ $\forall S_i \in \mathbf{S}$
    end for
end for
```

3.6 experiments

In this section, we first describe our dataset, followed by our experimental setup; comparative baselines, evaluation metrics, and implementation details. We then present results across several experiments to evaluate the performance of our model on merging semantically diverse induced-relations.

3.6.1 *Dataset*

We first evaluate our approach on multiple communities catering to different topics from a popular online Community Question Answer (CQA) platform, *StackExchange*[2]. The platform divides the communities into five different categories, i.e. Technology (**T**), Culture/Recreation (**C**), Life/Arts (**L**), Science (**S**) and Professional (**P**). For our analysis, we collect data from

[2] https://stackexchange.com/

the ten largest communities from each of the five categories until March 2019, resulting in a total of 50 StackExchange communities. The list of 50 StackExchange communities per category are;

TECHNOLOGY: AskUbuntu, Server Fault, Unix, TEX, Electronics, Gis, Apple, Wordpress, Drupal, DBA

CULTURE/RECREATION: English, Travel, RPG, Judaism, Puzzling, Bicycles, German, Christianity, BoardGames, History

LIFE/ARTS: Scifi, DIY, Academia, Graphic Design, Money, Photo, World-Building, Movies, Music, Law

SCIENCE: Stat, Physics, MathOverflow, CS, Chemistry, Biology, Philosophy, CS Theory, Economics, Astronomy

PROFESSIONAL/BUSINESS: Workplace, Aviation, Writers, Open source, Freelancing, CS Educators, Quant, PM, Parenting

In StackExchange, each questioner can mark a candidate answer as an "accepted" answer. We only consider questions with an accepted answer. Table 3.1 shows the final dataset statistics.

For each (q, a) tuple, we compute the following basic features:

ACTIVITY FEATURES: View count of the question, number of comments for both question and answer, the difference between posting time of question and answer, arrival rank of answer (we assign rank 1 to the first posted answer) (Tian, P. Zhang, and B. Li, 2013).

TEXT FEATURES: Paragraph and word count of question and answer, presence of code snippet in question and answer (useful for programming based forums), word count in the question title.

USER FEATURES: Word count in user profile's Aboutme section for both users; one who is posting the question and the other posting the answer.

Time-dependent features like upvotes/downvotes of the answer and user features like reputation or badges used in earlier studies on StackExchange (Burel, Mulholland, and Alani, 2016) are problematic for two reasons. First, we only know the aggregate values, not how these values change with time. Second, since these values typically increase over time, it is unclear if an accepted answer received the votes *prior* to or *after* an answer was accepted. Thus, we do not use such time-dependent features for our model and the baselines in our experiments.

	Technology			Culture/Recreation			Life/Arts			Science			Professional/Business		
	ServerFault	AskUbuntu	Unix	English	Games	Travel	SciFi	Home	Academia	Physics	Maths	Statistics	Workplace	Aviation	Writing
$\|Q\|$	61,873	41,192	9,207	30,616	12,946	6,782	14,974	8,022	6,442	23,932	18,464	13,773	8,118	4,663	2,932
$\|\mathcal{A}\|$	181,974	119,248	33,980	110,235	45,243	20,766	49,651	23,956	23,837	65,800	53,772	36,022	33,220	14,137	12,009
$\|U\|$	140,676	200,208	84,026	74,592	14,038	23,304	33,754	30,698	19,088	52,505	28,181	54,581	19,713	7,519	6,918
$\mu(\|\mathcal{A}_q\|)$	2.9	2.9	3.7	3.6	3.5	3.1	3.3	3.0	3.7	2.8	2.9	2.6	4.1	3.0	4.1

Table 3.1: Dataset statistics for the top three Stack Exchange communities from five different categories. $|Q|$: number of questions; $|\mathcal{A}|$: number of answers; $|U|$: number of users; $\mu(|\mathcal{A}_q|)$: mean number of answers per question. Professional/Business communities have slightly more answers per question on average than others. Technology communities are the largest in terms of number of question out of the five categories.

https://www.reddit.com/

REDDIT is another popular CQA platform with subreddits similar to StackExchange communities. In particular, we focus on Ask* subreddits as they are primarily used to seek help from a community of experts and non-experts. In particular, we crawled data from /r/askscience (science forum), /r/AskHistorians (history forum), and /r/AskDocs (medical forum) until October 2017. We performed basic preprocessing and removed posts or comments with single word/URLs or missing author/title information. We also removed infrequent users who posted less than two comments. Reddit has a hierarchical comment structure. For this work, we treat first-level comments as potential answers to the question. Users in these subreddits can get verified by providing anonymized verification documents including certification numbers, contact information, etc. to the moderators. We denote these verified users as experts. We treat an expert's comment as equivalent to an accepted answer and only consider posts which have an expert answer for our experiments. We discard posts with multiple experts' comment as it is hard to objectively choose a winner.

DATASET	$\|Q\|$	$\|\mathcal{A}\|$	$\|\mathcal{U}\|$	$\mu(\|\mathcal{A}_q\|)$
ASKDOCS	11,189	29,207	4,530	2.61
ASKHISTORIANS	15,425	45,586	11,761	2.96
ASKSCIENCE	37,990	121,278	32,117	3.19

Table 3.2: Dataset statistics for the Ask* Reddit communities. $|Q|$: number of questions; $|\mathcal{A}|$: number of answers; $|U|$: number of users; $\mu(|\mathcal{A}_q|)$: mean number of answers per question.

We employ 12 basic features for the Reddit dataset:
Activity features : ArrivalRank of the answer, number of subsequent comments on the answer, number of other answers to the question, Upvotes and downvotes for both, question and answer.
Text features : Word count of the question and answer

We employ post-vote features here as (Gilbert, 2013a) showed that there is widespread under-provision of voting on Reddit, partially due to long comment threads. It can act as a weak signal for answer quality. Unlike the StackExchange, Reddit voting is not biased by publicly visible acceptance of answers to a question. Thus, votes ideally represent the independent judgment of the crowd.

3.6.2 *Experimental Setup*

BASELINES We compare against state-of-the-art feature-based baselines for answer selection and competing aggregation approaches to fuse diverse relational views of the dataset (Schlichtkrull et al., 2018; Zhuang and Q. Ma, 2018).

RANDOM FOREST (RF) (Burel, Mulholland, and Alani, 2016; Tian, P. Zhang, and B. Li, 2013) model trains on the feature set mentioned earlier for each dataset. This model is shown to be the most effective feature-based model for Answer Selection.

FEED-FORWARD NETWORK (FF) (Jenders, Krestel, and Naumann, 2016) is used as a deep learning baseline to learn non-linear transformations of the feature vectors for each (q, a) tuple. This model is equivalent to our Reflexive GCN model in isolation.

DUAL GCN (DGCN) (Zhuang and Q. Ma, 2018) trains a separate GCN for each view. In addition to the supervised loss computed using training labels, they introduce a regularizer to minimize mean squared error (MSE) between vertex representations of two views, thus aligning the learned latent spaces. Formally, For instance,

$$\mathcal{L}_{reg}(Z_c, Z_{ts}) = \|\mathbf{Z}_c^K - \mathbf{Z}_{ts}^K\| \tag{3.22}$$

computes the MSE loss between Contrastive and TrueSkill Similarity GCN. Zhuang and Q. Ma (2018) proposed the model for two GCN representations and we extend it to four GCN with each GCN representing our relational view. The Contrastive view is seen to exhibit the best performance in isolation. Thus, the DualGCN loss can be given by:

$$\mathcal{L} = \mathcal{L}_0 + \lambda(t)\left(\sum_{S_i \in \mathbf{S}, S_i \neq c} \|\mathbf{Z}_c^K - \mathbf{Z}_i^K\|\right) \tag{3.23}$$

where $\mathcal{L}_0$ represents the supervised loss and $\mathbf{Z}_c^K$ is the vertex representations of the Contrastive GCN. The regularizer loss is similar to our intra-relation aggregation approach but assumes label and feature sharing across *all* the views.

RELATIONAL GCN (RGCN) (Schlichtkrull et al., 2018) combines the output representations of previous layer of each view to compute an aggregated input to the current layer, i.e., $\mathbf{Z}_i^{k-1}$ of layer $k-1$ of each view is used to compute an aggregated input to layer k. Formally,

$$\mathbf{Z}_{rgcn}^{k} = \sigma\left(\sum_{S_i \in \mathbf{S}} \mathbf{Z}_i^{k-1}\right) \tag{3.24}$$

where Z_{rgcn} is final output of this model at layer k and σ is the activation function.

Relational GCN (RGCN) (Schlichtkrull et al., 2018) combines the output representations of previous layer of each view to compute an aggregated input to the current layer, i.e., $\mathbf{Z}_i^{k-1}$ of layer $k-1$ of each view is used to compute an aggregated input to layer k. Formally,

We also report results for each view individually: Contrastive (C-GCN), Arrival Similarity (AS-GCN), TrueSkill Similarity (TS-GCN), and Reflexive (R-GCN) with our proposed IR-GCN model. We do not compare with other graph structure-based approaches to compute vertex representations (Grover and Leskovec, 2016; Perozzi, Al-Rfou, and Skiena, 2014; Jian Tang et al., 2015; Z. Yang, Cohen, and Salakhutdinov, 2016) as GCN is shown to outperform them (Kipf and Welling, 2016a). We also later compare with common aggregation strategies to merge neural representations discussed earlier in section 3.5.

EVALUATION METRIC We randomly select 20% of the questions, $\mathbf{T}_q \subset Q$ to be in the test set. Then, subsequently all (q, a) tuples such that $q \in \mathbf{T}_q$ comprise the set of test tuples or vertices, $\mathbf{T}$. The rest of the vertices, along with their label information, is used for training the model. We evaluate our model on two metrics, Accuracy and Mean Reciprocal Rank (MRR). Accuracy metric is widely used in vertex classification literature while MRR is popular for ranking problems like answer selection. Formally,

$$Acc = \frac{1}{|\mathbf{T}|} \sum_{(q,a) \in \mathbf{T}} \mathbb{1}\left(y_{(q,a)} \cdot h_b((q,a)) > 0\right) \tag{3.25}$$

with $\cdot$ as the product and $\mathbb{1}(.)$ as the indicator function. The product is positive if the accepted label and predicted label match and negative otherwise.

$$MRR = \frac{1}{|\mathbf{T}_q|} \sum_{q \in \mathbf{T}_q} \frac{1}{\sum_{a' \in \mathcal{A}(q)} \mathbb{1}\left(L_{(q,a)} < L_{(q,a')}\right)} \tag{3.26}$$

where $L_{(q,a)}$ refers to the position of accepted answer a in the ranked list for question q (X.-J. Wang et al., 2009).

implementation details We implemented our model and the baselines in Pytorch. We use ADAM optimizer (Kingma and Ba, 2014) for training with 50% dropout to avoid overfitting. We use four hidden layers in each GCN with hidden dimensions 50, 10, 10, 5, respectively, and ReLU activation. The coefficients of $\mathcal{L}_1$ and $\mathcal{L}_2$ regularizers are set to $\gamma_1 = 0.05$ and $\gamma_2 = 0.01$ respectively. For TrueSkill Similarity, we use margin $\delta = 4$ to create links, while for Arrival similarity, we use $\delta = 0.95$. We implement a mini-batch version of training for large graphs where each batch contains a set of questions and their associated answers. This mini-batch version is equivalent to training on the whole graph as we have disconnected cliques.

3.6.3 *Performance Analysis*

Table 3.3 shows impressive gains over state-of-the-art baselines for all the five categories of StackExchange. We report mean results for each category obtained after 5-fold cross-validation on each of the communities. Our induced-relational GCN model beats best performing baseline by 4-5% on average in accuracy. The improvement in MRR values is around 2.5-3% across all categories. Note that MRR is based only on the rank of the accepted answer, while accuracy is based on correct labeling of *both* accepted and non-accepted answers.

Among individual views, Contrastive GCN performs best on all the communities. It even beats the best performing baseline DualGCN that uses all the relational views. Note that the contrastive view compares between the candidate answers to a question and uses our proposed contrastive modification to the convolution operation. Arrival Similarity follows Contrastive and then Reflexive. The superior performance of the Arrival Similarity view shows that early answers tend to get accepted and vice versa. It indicates that users primarily use CQA forums for quick answers to their queries. Also, recall that Reflexive predicts each vertex's label independent of other answers to the same question. Thus, the competitive performance of the Reflexive strategy indicates that vertex's features itself are well predictive of the label. TrueSkill Similarity performs at par or slightly worse than Reflexive. Figure 3.5 presents t-SNE distributions (Maaten and Hinton, 2008) of the learned vertex representations ($\mathbf{Z}_i^K$) of our model applied to Chemistry StackExchange from Science category. Note that each view, including two views under Similar Contrast relation, learns a distinct vertex

Method	Technology		Culture/Recreation		Life/Arts		Science		Professional/Business	
	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF (Burel, Mulholland, and Alani, 2016; Tian, P. Zhang, and B. Li, 2013)	66.780 (23)	0.683 (43)	72.500 (18)	0.626 (50)	72.710 (49)	0.628 (89)	68.090 (24)	0.692 (49)	74.720 (44)	0.595 (81)
FF (Jenders, Krestel, and Naumann, 2016)	67.310 (27)	0.786 (22)	72.220 (20)	**0.782 (23)**	73.580 (49)	0.780 (34)	67.870 (24)	0.800 (28)	74.630 (40)	0.759 (49)
DGCN (Zhuang and Q. Ma, 2018)	70.700 (22)	0.782 (17)	75.220 (17)	0.771 (28)	76.730 (34)	0.784 (38)	**71.450 (23)**	0.791 (35)	76.860 (31)	0.751 (46)
RGCN (Schlichtkrull et al., 2018)	54.400 (45)	0.673 (45)	60.390 (16)	0.645 (42)	59.970 (43)	0.654 (54)	58.650 (54)	0.682 (42)	63.020 (38)	0.657 (61)
AS-GCN	67.760 (32)	0.775 (15)	73.050 (21)	0.763 (25)	73.790 (48)	0.776 (42)	66.930 (45)	0.788 (28)	74.990 (45)	0.742 (47)
TS-GCN	66.870 (32)	0.779 (18)	72.160 (23)	0.764 (23)	72.020 (61)	0.765 (48)	65.900 (42)	0.790 (31)	74.170 (46)	0.747 (44)
C-GCN	**71.640 (22)**	**0.790 (15)**	**76.180 (17)**	0.781 (24)	**77.370 (34)**	**0.788 (40)**	70.810 (42)	**0.800 (32)**	**77.570 (38)**	**0.768 (34)**
IR-GCN	73.960 (23)	0.794 (14)	78.610 (18)	0.790 (25)	79.210 (32)	0.800 (37)	**74.980 (21)**	**0.808 (28)**	**80.170 (26)**	**0.785 (32)**

Table 3.3: DGCN stands for DualGCN, RGCN stands for RelationalGCN, and IR-GCN stands for Induced Relational GCN. Accuracy and MRR values for StackExchange with state-of-the-art baselines. Our model outperforms by at least 4% in Accuracy and 2.5% in MRR. Contrastive GCN performs best among individual views. The model results in bold the *second-best* performance among all other models. Our model shows statistical significance at level $p < 0.01$ over all second best model on single tail paired t-test.

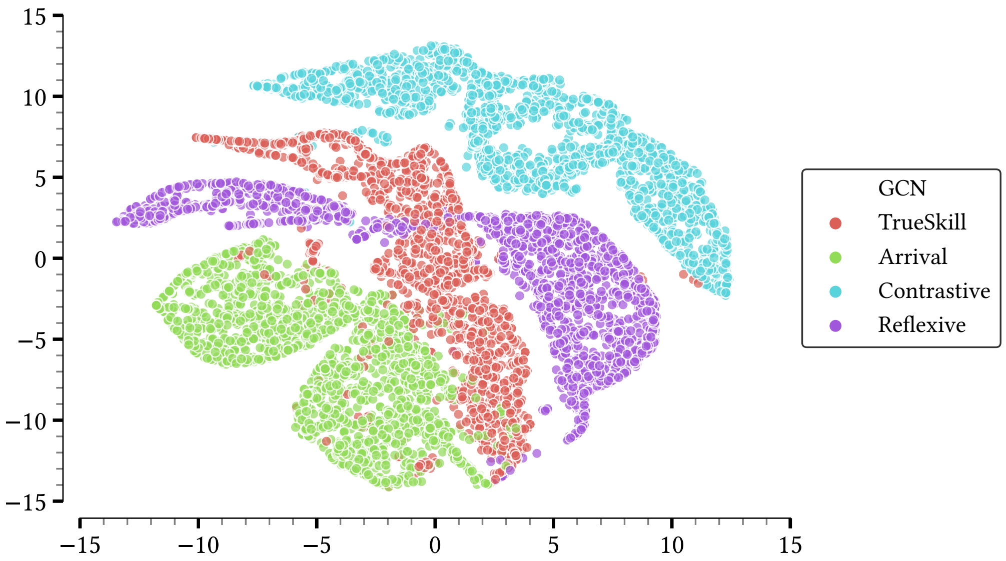


Figure 3.5: t-stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) distributions of the learned vertex representations by our model for Chemistry StackExchange. Each view learns a distinct vertex representation. Best viewed in color.

representation. Hence, all views are essential and contribute to our final performance.

Out of the baseline graph ensemble approaches, DualGCN performs significantly better than RelationalGCN by an average of around 26% for all categories. Recall that in the RelationalGCN model, the convolution output of each view is linearly combined to compute the final output. Linear combination works well for knowledge graphs as each view can be thought of as a feature, and then it accumulates information from each feature. DualGCN is similar to our approach and trains different GCN for each view and later merges their results. However, it enforces similarity in vertex representations learned by each view. This restriction is not suitable for our induced-relationships as they are semantically different (contrastive captures contrast in features vs. similarity enforces label sharing).

Table 3.4 shows performance gains over the state-of-art baselines for the Reddit dataset. All results are reported after 5-fold cross-validation. Our model improves by 16% on average in accuracy over the baselines for Reddit. The improvement in MRR is at an average increase of 7% than the baseline higher than for StackExchange.

Among individual views, for Reddit, there is a considerable difference in performance for each view. TrueSkill Similarity performs much better, followed by Arrival Similarity and Contrastive. Reflexive GCN performs

Method	AskDocs		AskHistorians		AskScience	
	Acc (%)	MRR	Acc(%)	MRR	Acc(%)	MRR
RF (Burel, Mulholland, and Alani, 2016; Tian, P. Zhang, and B. Li, 2013)	59.35	0.70	65.62	0.71	65.87	0.71
FF (Jenders, Krestel, and Naumann, 2016)	62.30	0.72	67.89	0.73	68.99	0.71
DGCN (Zhuang and Q. Ma, 2018)	77.54	0.79	80.49	0.81	75.57	0.82
RGCN (Schlichtkrull et al., 2018)	57.98	0.67	64.56	0.68	62.42	0.64
AS-GCN	76.53	0.79	80.70	0.78	78.14	0.80
TS-GCN	84.44	0.86	90.95	0.83	87.61	0.82
C-GCN	67.39	0.75	70.57	0.74	71.11	0.77
IR-GCN	**87.60**	**0.90**	**93.81**	**0.85**	**89.11**	**0.84**

Table 3.4: Accuracy and MRR values for Ask Reddits. Our model significantly outperforms by 16% in Accuracy and 7% in MRR. TrueSkill Similarity performs best among individual IR-GCNs.

the worst for Reddit as it predicts each node's label independent of answers to the same question.

Out of the baseline graph ensemble approaches, DualGCN and RelationalGCN, similar to StackExchange, DualGCN consistently performs better than RelationalGCN by an average of around 3% for Reddit.

3.7 DISCUSSION

In this section, we first evaluate the importance of each relational view for our boosted model. We then compare with approaches proposed to merge neural networks in general in other domains. We then illustrate *discriminative magnification effect* in detail and study the robustness of our model to training label sparsity. We also extend our proposed approach to include textual features and compare it with a text-based model. Finally, we provide a theoretical analysis of performance gains of our Contrastive GCN model and provide limitations of our approach.

3.7.1 *Ablation Study on Relation Types*

Relation Type	Tech	Culture	Life	Sci	Business	AskDocs
C	71.23	75.90	78.71	72.99	76.85	67.39
{ TS, AS }	67.86	74.15	75.75	65.80	76.13	84.57
R	68.30	73.35	76.57	67.40	75.76	62.30
{TS, AS } + R	69.28	75.50	76.41	70.11	77.90	86.34
C + R	73.04	77.66	80.25	73.72	80.04	70.02
C + { TS, AS }	72.81	78.04	81.41	72.19	80.15	86.99
C + { TS, AS } + R	**73.87**	**78.74**	**81.60**	**74.68**	**80.56**	**87.60**

Table 3.5: 5-fold Accuracy (in %) comparison for different combination of relation types for our boosted model. Contrastive and Similar Contrast relations together performs similar to the final model.

We present results of an ablation study with different combination of relation types (Contrastive, Similar and Reflexive) used for IR-GCN model in Table 3.5. We conducted this study on the biggest community from each of the five categories, i.e., ServerFault (Technology), English (Culture), Science Fiction (Life), Physics (Science), Workplace (Business). We also report results for AskDocs subreddit. Similar Contrast relation (TrueSkill and Arrival) used in isolation perform the worst among all the variants. Training Contrastive and Similar Contrast relation together in our boosted framework performs similar to our final model. Reflexive GCN contributes the least as it does not consider any neighbors.

3.7.2 *Aggregator Architecture Variants*

We compare our gradient boosting based aggregation approach with other popular methods used in literature to merge different neural networks discussed in section 3.5.

Table 3.6 reports the accuracy results for these aggregator variants as compared to our model. Our method outperforms all the variants with Fusion performing the best. This superior performance affirms that existing

Method	Tech	Culture	Life	Sci	Business	AskDocs
Stacking (Yu, Yin, and Zhu, 2017)	68.58	74.44	79.19	70.29	75.50	85.40
Fusion (Farnadi et al., 2018)	72.30	77.25	80.79	73.91	79.01	86.33
Neighbor Agg (Hamilton, Z. Ying, and Leskovec, 2017; Schlichtkrull et al., 2018)	69.29	74.28	77.94	68.42	78.64	86.00
IR-GCN	**73.87**	**78.74**	**81.60**	**78.00**	**80.56**	**87.60**

Table 3.6: 5-fold Accuracy (in %) comparison of different aggregator architectures. These architectures perform worse than Contrastive GCN for StackExchange. Fusion performs similarly but is computationally expensive.

aggregation models are not suitable for our problem. Note that these approaches perform worse than even Contrastive GCN except Fusion. The fusion approach performs similarly to our approach but is computationally expensive as the input size for each view is linear in the number of all views in the model.

3.7.3 *Discriminative Magnification effect*

We show that due to our proposed modification to the convolution operation for contrastive view, we achieve *Discriminative Magnification effect* (eq. (3.11)). Note that the difference is scaled by Clique size $(1 + 1/n - 1)$, i.e. number of answers to a question, $|\mathcal{A}_q|$. Figure 3.6 shows the accuracy of our IR-GCN model as compared to the FeedForward model with varying clique size. Recall that the FeedForward model predicts node labels independent of other nodes and is not affected by clique size. We report average results over the same five communities as above. We can observe that increase in accuracy is much more for lower clique sizes (13% improvement for $|\mathcal{A}_q| = 2$ and 4% for $|\mathcal{A}_q| = 3$ on average). The results are almost similar for larger clique sizes. In other words, our model significantly outperforms the FeedForward model for questions with fewer candidate answers. However,

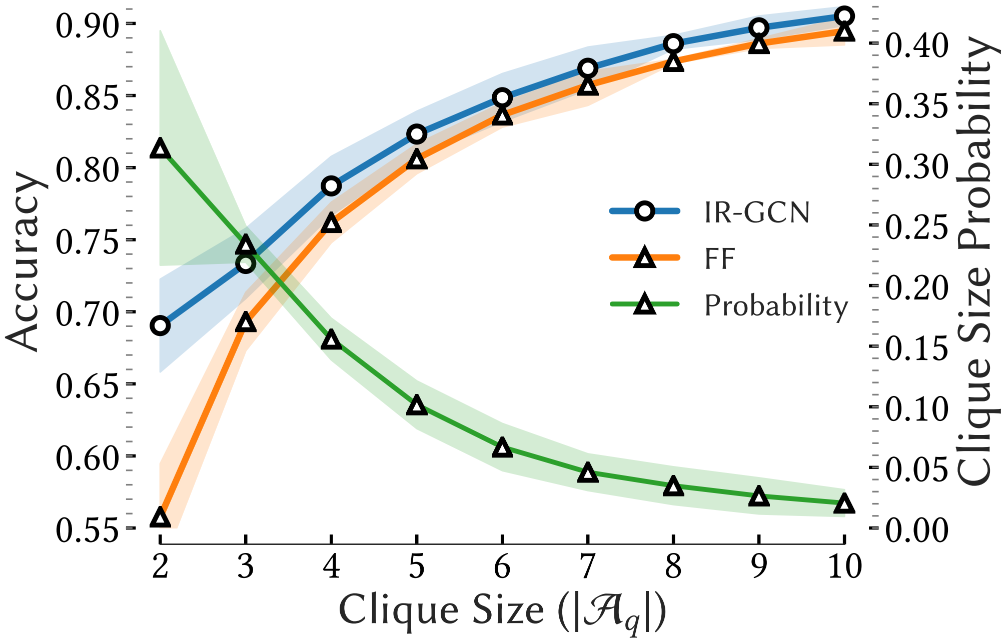


Figure 3.6: Accuracy of our IR-GCN model compared to the FF model with varying clique size (i.e. number of answers to a question, $|\mathcal{A}_q|$) for Contrastive view . We report averaged results over the largest community of all categories. Our model performs much better for smaller cliques, and the effect diminishes for larger cliques (eq. (3.11)). 80% of the questions have < 4 answers.

around 80% of the questions have very few answers(< 4), and thus this gain over FF is significant.

Alternatively, we also plot the probability of error per tuple given each clique size ($p(e|k)$) for the movie StackExchange in Figure 3.7. The *standard* corresponds to a naive baseline of randomly selecting an accepted answer within each clique. For this standard baseline, error probability per clique can be denoted as,

$$p(e|k) = (1 - \frac{1}{k}) \times \frac{2}{k} = 2\frac{(k-1)}{k^2} \tag{3.27}$$

$(1 - 1/k)$ denotes the probability of choosing the wrong accepted answer, while $2/k$ is the actual error rate in these scenarios. The error rate is such because even in cases where the baseline chose the wrong accepted answer, remaining answers are still correctly classified as not accepted. Thus, there are only two errors per clique.

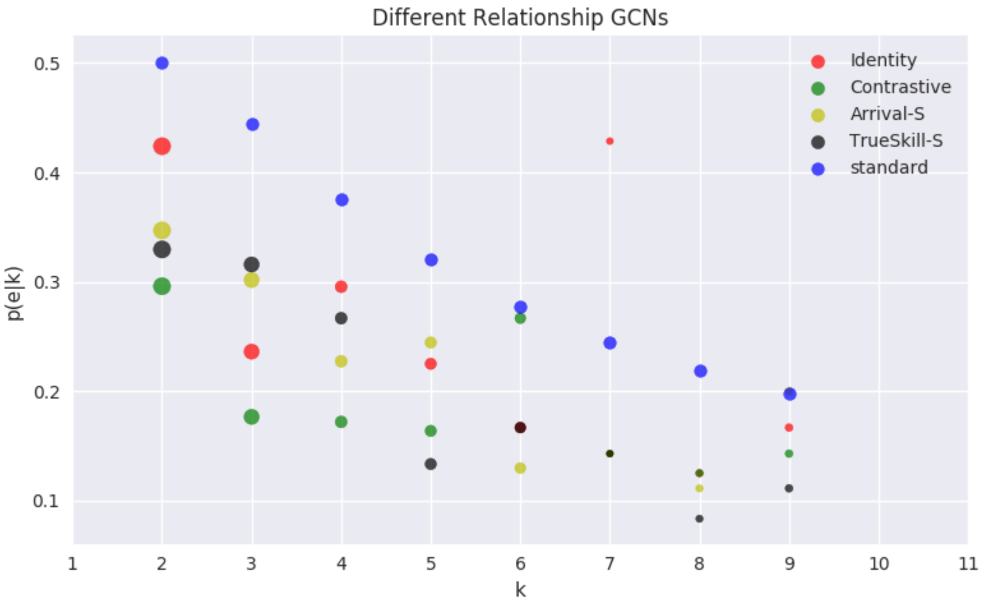


Figure 3.7: Probability of error with varying clique size for movie StackExchange. Standard represents random selection. Contrastive view outperforms other views for smaller clique sizes.

The standard baseline performs the worst as the error probability is highest than the other baselines for each clique. The Contrastive view has the least error probability for smaller cliques ($k < 5$). This result is analogous to the performance gain illustrated above due to the *Discriminative Magnification effect*. For larger cliques, similar contrast views (ArrivalSkill and TrueSkill) have the least error probability. As both of these views connects similar tuples across different questions, they are thus more useful for questions with a higher number of competing answers.

3.7.4 *Label Sparsity*

Graph Convolution Networks are robust to label sparsity as they exploit graph structure and are thus heavily used for semi-supervised settings. Figure 3.8 shows the change in accuracy for Physics StackExchange from the Science category at different training label rates. Even though our graph contains disconnected cliques, IR-GCN still preserves robustness to label sparsity. In contrast, the accuracy of the FeedForward model declines sharply with less label information. Performance of DualGCN remains relatively stable while Relational GCN's performance increases with a decrease in label rate. Relational GCN assumes each view to be of similarity

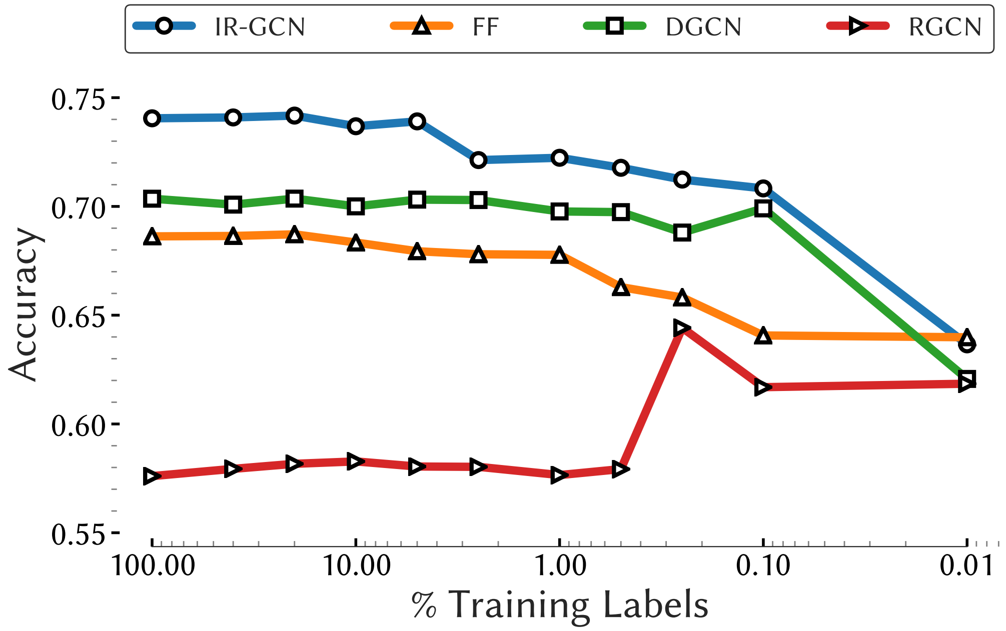


Figure 3.8: Change in accuracy with varying training label rates for Physics StackExchange. Our model is more robust to label sparsity than other relation ensemble approaches. RGCN works better with fewer labels as contrastive relation introduces noise in the model. At extreme sparsity, all approaches converge to the same value indicating random selection.

relation, and thus, adding contrastive relation introduces noise in the model. However, as the training labels become extremely sparse, the training noise decreases that leads to a marked improvement in the model. In the case of an extremely low label rate of 0.01%, all approaches converge to the same value, which is the expectation of theoretically random selection. We also obtained similar results for the other four StackExchange communities.

3.7.5 *Including Textual Features*

Most of the current literature focuses on using textual similarity for Answer Selection. In this section, we compare our proposed IR-GCN model to a popular text-based model (Tan, Xiang, and B. Zhou, 2015) for answer selection.

TEXT PREPROCESSING: For this experiment, we first preprocessed the text of both questions and answers. We first removed all code snippets, HTML tags, stopwords, and URLs from the text of all questions and answers. We then tokenized the text using NLTK tokenizer followed by lemmatization using WordNetLemmatizer and finally converted it into lowercase.

We use `torchtext` (https://pytorch.org/text/) to create vocabulary and limit the text of each question and answer to be 250 words long. We initialized the words in the vocabulary using 300-dimensional pre-trained embeddings from `Word2vec` (https://code.google.com/archive/p/word2vec/). We randomly initialized words present in the vocabulary but not in word2vec.

We evaluate multiple approaches to test the effectiveness of incorporating textual features for answer selection task.

QA-LSTM/CNN (Tan, Xiang, and B. Zhou, 2015) uses a stacked bidirectional LSTM model followed by convolution filters to extract embeddings for the question and answer text separately. Answers are then classified according to the cosine similarity of learned question and answer embedding.

Specifically, in this baseline, we use a biLSTM model with a hidden dimension = 300, followed by 50 1D convolutional filters with a kernel size of 3. We then compute the final embeddings by applying 1D max-pooling on the output of the convolution layer. We also used Tanh nonlinearity and a dropout of 0.3 on the final embeddings. We finally use these embeddings to compute a cosine similarity score between a question and its answers. This score is used to rank the candidate answers for evaluation. We implemented the baseline in Pytorch.

Textual Similarity (T-GCN): We create a *SimilarContrast* view that connects answers authored by a user where her answer is significantly similar (dissimilar) to the question than other competing answers. We used cosine similarity on the learned question and answer embedding from the QA-LSTM/CNN approach as the similarity function.

Specifically, we extract the updated embeddings of the question and answer text from the learnt QA-LSTM model. We then compute cosine similarity between the embeddings of each question and its answers. We then connect answers authored by a specific user, where the difference in cosine similarity of the answer with the other competing answers is greater than margin λ. Specifically, if the user authors answers a, a' to questions q, q', we create a link between a and a' if

$$|C_{q,a} - C_{q,b}| > \lambda; \forall b \in \mathcal{A}_q \tag{3.28}$$

$$|C_{q,a'} - C_{q,c}| > \lambda; \forall c \in \mathcal{A}_{q'} \tag{3.29}$$

where $C_{q,a}$ is the cosine similarity of the answer a with respect to question q. Similarly, a link is created for the opposite case when difference is less than $-\lambda$. In our experiments, we assign $\lambda = 0.4$. The hypothesis is that irrelevant(dissimilar) answers will more likely be rejected and vice versa.

IR-GCN + T-GCN extends our proposed model to also include the Textual Similarity as the third *SimilarContrast* view in addition to Arrival and TrueSkill Similarity.

Method	Tech	Culture	Life	Sci	Business
QA-LSTM/CNN (Tan, Xiang, and B. Zhou, 2015)	66.49	71.70	69.42	62.91	72.55
FF (Jenders, Krestel, and Naumann, 2016)	68.30	73.35	76.57	67.40	75.76
C-GCN	71.23	75.90	78.71	72.99	76.85
T-GCN	69.25	73.77	76.39	67.79	77.08
IR-GCN	73.87	78.74	81.60	74.68	80.56
IR-GCN + T-GCN	73.89	78.00	81.07	74.49	78.86

Table 3.7: 5-fold Accuracy comparison of text-based baseline and textual similarity GCN with IR-GCN.

In general, the text-based baseline, QA-LSTM, performs worse than even reflexive GCN, as shown in Table 3.7. Note that reflexive GCN employs a feedforward model on the activity and user features used in our experiments. This worse performance is surprising as most of the current literature focuses on textual features for the task. Our results indicate that non-textual features are useful too for answer selection task on StackExchange communities.

Textual Similarity GCN performs better than QA-LSTM and Reflexive GCN. Even though we use the output of QA-LSTM to construct the graph for T-GCN, the graph improves performance as it connects answers across questions. However, adding the T-GCN view in our proposed IR-GCN model decreases the performance slightly. One possible explanation could be that similar contrast views based on user features (Arrival similarity and TrueSkill similarity) are not compatible with views based on textual features.

We further replaced our activity-based features with the learned embeddings obtained after training the QA-LSTM/CNN (Tan, Xiang, and B. Zhou, 2015) model as the node features. We observed that the performance of all approaches went down slightly when using textual features only (Table 3.8). As we noted before, GCNs aggregate features among the

Method	Tech	Culture	Life	Sci	Business
QA-LSTM/CNN (Tan, Xiang, and B. Zhou, 2015)	66.49	71.70	69.42	62.91	72.55
FF (Jenders, Krestel, and Naumann, 2016)	66.00	72.22	69.85	63.63	75.57
C-GCN	66.19	72.45	70.23	63.89	75.71
CT-GCN	66.06	72.35	71.88	64.14	75.69
IR-GCN	66.56	72.92	72.54	65.11	75.95
IR-GCN + T-GCN	66.49	73.17	72.85	65.29	75.86

Table 3.8: 5-fold Accuracy comparison of text-based baseline and textual similarity GCN with learnt text embeddings as features in the GCN.

neighbors. In our similar contrast views, it is not favorable to aggregate textual features among the neighbors as it connects answers catering to different questions. Thus, aggregating textual features creates noise in the model leading to worse performance [3].

3.7.6 *Contrastive GCN Analysis*

The ability of neural networks to perform classification in sparse high-dimensional manifolds has been studied in past work, especially in the context of adversarial learning (Lu, Issaranon, and Forsyth, 2017). We employ the ReLU activation function in our convolution layers and study the outputs of the kth layer, i.e., embeddings with k-order locality. This transformation breaks the input space into cells with smooth gradients within each cell, at whose boundaries the piecewise linear function changes (i.e., the likelihood of the two classes of answers).

We ask a specific question in the context of our Contrastive GCN. *What is the impact of the layerwise discriminative magnification induced by our formulation?* Discriminative magnifications result in improved separability of the two classes in the later convolving layers, an effect we earlier demonstrated with a sample network in fig. 3.3. This positively impacts the ability of the model to explain the observed data points (i.e., create p-domains that are well aligned with the contrastive samples provided) and improve the

[3] We also experimented with concatenating textual features with the original features used in the previous experiments. However, the performance was still a little worse than the results with only original features.

generalizability of the learned model to unseen data points. However, it is crucial to maintain sufficient regularization with weight decay to prevent sparse regions exhibiting sharp gradients that could affect model performance.

The capacity of our model can also be quantified in terms of the VC dimension of the aggregated classifier against the individual learners. Gradient boosting with multiple relation learners (each of which captures a specific aspect of node locality via graph convolution on the induced relations) could boost the capacity of the joint model, enabling better generalization and a more accurate fit in the data manifold (i.e., higher capacity to fit regions to fine distinctions).

Let us denote the upper bound of the VC dimension or capacity of each individual learner as D (If the individual learners do not have identical capacity, the minimum can be used to compute a lower bound on the aggregated learner capacity). Then the gradient boosted learner with T classifiers has a bound on it's capacity (Shalev-Shwartz and Ben-David, 2014) given by,

$$\mathcal{VC}_{Agg} = T \times (D+1) \times (3\log(T.(D+1)) + 2) \tag{3.30}$$

Thus we identify two potential reasons for our performance gains, first the discriminative magnification effect that also supports the strong individual performance of the contrast view, and second the gain in capacity from boosting, which could explain its advantage over competing aggregation methods.

3.7.7 *Limitations*

We do recognize certain limitations of our work. First, we focus on equivalence relations that induce a graph comprising cliques. While cliques are useful graph objects for answer selection, equivalence relations may be too restrictive for other problems (e.g., the relation is not transitive). However, our modular framework does apply to arbitrary graphs, except that Equation (3.7) will no longer be an *exact* convolution but be an approximation. Second, we assume no evolution in author skills. This assumption is not correct as users evolve with experience. We aim to address this in future work.

In summary, our model showed significant gains over state-of-the-art baselines for combining information from semantically different relational links in a graph. Our model is also more robust to training label sparsity as compared to other aggregator GCN approaches. We reasoned that the performance gains achieved by our aggregation strategy could be attributed in part to the enhanced learning capacity of the boosted model and the effect of discriminative feature magnification. We showed that content can also be used to induce graphs and performs better than using content features in isolation. Finally, we presented a few limitations and possible future extensions.

3.8 CONCLUSION

This chapter addressed the question of identifying the accepted answer to a question in CQA forums. We developed a novel induced relational graph convolutional (IR-GCN) framework to address this question. We made three contributions. First, we introduced a novel idea of using strategies to induce different views on (q, a) tuples in CQA forums. Each view consists of cliques and encodes—reflexive, similar, contrastive—relation types. Second, we encoded label sharing and label contrast mechanisms within each clique through a GCN architecture. Our novel contrastive architecture achieves *Discriminative Magnification* between nodes. Finally, we show through extensive empirical results on StackExchange that boosting techniques improved learning in our convolutional model. This was a surprising result since much of the work on neural architecture that are strong learners focuses on stacking, fusion or aggregator architectures. However, boosting is traditionally shown to be most effective with weak learners. Our ablation studies show that the contrastive relation is most effective individually in StackExchange.

Part III

APPENDIX

A
APPENDIX TEST

Lorem ipsum at nusquam appellantur his, ut eos erant homero concludaturque. Albucius appellantur deterruisset id eam, vivendum partiendo dissentiet ei ius. Vis melius facilisis ea, sea id convenire referrentur, takimata adolescens ex duo. Ei harum argumentum per. Eam vidit exerci appetere ad, ut vel zzril intellegam interpretaris.

More dummy text.

A.1 APPENDIX SECTION TEST

Test: Table A.1 (This reference should have a lowercase, small caps A if the option `floatperchapter` is activated, just as in the table itself → however, this does not work at the moment.)

LABITUR BONORUM PRI NO	QUE VISTA	HUMAN
fastidii ea ius	germano	demonstratea
suscipit instructior	titulo	personas
quaestio philosophia	facto	demonstrated

Table A.1: Autem usu id.

A.2 ANOTHER APPENDIX SECTION TEST

Equidem detraxit cu nam, vix eu delenit periculis. Eos ut vero constituto, no vidit propriae complectitur sea. Diceret nonummy in has, no qui eligendi recteque consetetur. Mel eu dictas suscipiantur, et sed placerat oporteat. At ipsum electram mei, ad aeque atomorum mea. There is also a useless Pascal listing below: Listing A.1.

Listing A.1: A floating example (`listings` manual)

```
for i:=maxint downto 0 do
begin
{ do nothing }
end;
```

BIBLIOGRAPHY

Adamic, Lada A., Jun Zhang, Eytan Bakshy, and Mark S. Ackerman (2008). "Knowledge Sharing and Yahoo Answers: Everyone Knows Something." In: WWW'08. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367587 (cit. on p. 13).

Agichtein, Eugene, Eric Brill, Susan Dumais, and Robert Ragno (2006). "Learning user interaction models for predicting web search result preferences." In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–10 (cit. on p. 1).

Agrawal, Sweta and Amit Awekar (2018). "Deep learning for detecting cyberbullying across multiple social media platforms." In: *European Conference on Information Retrieval*. Springer, pp. 141–153 (cit. on p. 19).

Angeletou, Sofia, Matthew Rowe, and Harith Alani (2011). "Modelling and Analysis of User Behaviour in Online Communities." In: *The Semantic Web-ISWC'11*. ISBN: 978-3-642-25073-6 (cit. on p. 13).

Badjatiya, Pinkesh, Shashank Gupta, Manish Gupta, and Vasudeva Varma (2017). "Deep learning for hate speech detection in tweets." In: *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, pp. 759–760 (cit. on p. 19).

Barrón-Cedeno, Alberto, Simone Filice, Giovanni Da San Martino, Shafiq R Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti (2015). "Thread-Level Information for Comment Classification in Community Question Answering." In: *ACL (2)*, pp. 687–693 (cit. on p. 17).

Benevenuto, Fabricio, Tiago Rodrigues, Meeyoung Cha, and Virgilio Almeida (2009). "Characterizing User Behavior in Online Social Networks." In: IMC'09. ISBN: 978-1-60558-771-4. DOI: 10.1145/1644893.1644900 (cit. on p. 13).

Berg, Rianne van den, Thomas N. Kipf, and Max Welling (2017). "Graph Convolutional Matrix Completion." In: *CoRR* abs/1706.02263 (cit. on p. 20).

Beutel, Alex (2016). "User behavior modeling with large-scale graph analysis." In: *Computer Science Department, Carnegie Mellon University* (cit. on p. 1).

Bicego, Manuele, Vittorio Murino, and Mário A. T. Figueiredo (2003). "Similarity-based Clustering of Sequences Using Hidden Markov Models." In: *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer-Verlag, pp. 86–95. ISBN: 3-540-40504-6 (cit. on p. 13).

Biryukov, Maria and Cailing Dong (2010). "Analysis of Computer Science Communities Based on DBLP." In: *Proceedings of the 14th European Conference on Research and Advanced Technology for Digital Libraries*. Springer-Verlag. ISBN: 3-642-15463-8, 978-3-642-15463-8 (cit. on p. 16).

Brugere, Ivan, Brian Gallagher, and Tanya Y. Berger-Wolf (2018). "Network Structure Inference, A Survey: Motivations, Methods, and Applications." In: *ACM Comput. Surv.* DOI: 10.1145/3154524 (cit. on p. 27).

Burel, Grégoire, Paul Mulholland, and Harith Alani (2016). "Structural Normalisation Methods for Improving Best Answer Identification in Question Answering Communities." In: *International Conference on World Wide Web, WWW* (cit. on pp. 17, 25, 27, 30, 44, 47, 50, 52).

Burges, Christopher J. C. (2010). *From RankNet to LambdaRank to LambdaMART: An Overview*. Tech. rep. Microsoft Research (cit. on p. 25).

Burges, Christopher J., Robert Ragno, and Quoc V. Le (2007). "Learning to Rank with Nonsmooth Cost Functions." In: *Advances in Neural Information Processing Systems*. MIT Press (cit. on p. 25).

Cai, Chenwei, Ruining He, and Julian McAuley (2017). "SPMC: Socially-Aware Personalized Markov Chains for Sparse Sequential Recommendation." In: *CoRR* abs/1708.04497. arXiv: 1708.04497 (cit. on pp. 14, 15).

Chakraborty, Tanmoy and Subrata Nandi (Feb. 2018). "Universal Trajectories of Scientific Success." In: *Knowl. Inf. Syst.* 54.2, pp. 487–509. ISSN: 0219-1377. DOI: 10.1007/s10115-017-1080-y (cit. on p. 16).

Cheng, Chen, Haiqin Yang, Michael R. Lyu, and Irwin King (2013). "Where You Like to Go Next: Successive Point-of-interest Recommendation." In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, pp. 2605–2611. ISBN: 978-1-57735-633-2 (cit. on p. 14).

Chung, Fan R. K. (1997). *Spectral graph theory*. Vol. 92. CBMS Regional Conference Series in Mathematics. American Mathematical Society (cit. on p. 33).

Clauset, Aaron, Samuel Arbesman, and Daniel B Larremore (2015). "Systematic inequality and hierarchy in faculty hiring networks." In: *Science advances* 1.1, e1400005 (cit. on p. 16).

Coviello, Emanuele, Antoni B. Chan, and Gert R. G. Lanckriet (2014). "Clustering Hidden Markov Models with Variational HEM." In: *J. Mach. Learn. Res.* 15, pp. 697–747. ISSN: 1532-4435 (cit. on p. 13).

Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber (2017). "Automated hate speech detection and the problem of offensive language." In: *Eleventh international aaai conference on web and social media* (cit. on p. 19).

De Choudhury, Munmun, Emre Kiciman, Mark Dredze, Glen Coppersmith, and Mrinal Kumar (2016). "Discovering Shifts to Suicidal Ideation from Mental Health Content in Social Media." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, pp. 2098–2110. ISBN: 9781450333627. DOI: `10.1145/2858036.2858207` (cit. on p. 4).

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering." In: *Advances in Neural Information Processing Systems* (cit. on pp. 20, 33, 34).

Derr, Tyler, Yao Ma, and Jiliang Tang (2018). "Signed Graph Convolutional Network." In: *CoRR* (cit. on pp. 20, 26, 27, 30, 38).

Deville, Pierre, Dashun Wang, Roberta Sinatra, Chaoming Song, Vincent D Blondel, and Albert-László Barabási (2014). "Career on the move: Geography, stratification, and scientific impact." In: *Scientific reports* 4, p. 4770 (cit. on p. 16).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "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)*, pp. 4171–4186 (cit. on p. 1).

Duvenaud, David K., Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams (2015). "Convolutional Networks on Graphs for Learning Molecular Fingerprints." In: *Advances in Neural Information Processing Systems* (cit. on p. 20).

ElSherief, Mai, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding (2018). "Hate lingo: A target-based linguistic analysis of hate speech in social media." In: *Twelfth International AAAI Conference on Web and Social Media* (cit. on p. 6).

Fan, Wenqi, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin (2019). *Graph Neural Networks for Social Recommendation*. arXiv: `1902.07243 [cs.IR]` (cit. on p. 21).

Farnadi, Golnoosh, Jie Tang, Martine De Cock, and Marie-Francine Moens (2018). "User Profiling Through Deep Multimodal Fusion." In: *International Conference on Web Search and Data Mining*. WSDM '18. ACM (cit. on pp. 39, 54).

Feng, Minwei, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou (2015). "Applying deep learning to answer selection: A study and an open task." In: *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU* (cit. on p. 18).

Freund, Yoav and Robert E. Schapire (1995). "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting." In: *European Conference on Computational Learning Theory*. Springer-Verlag (cit. on p. 40).

Furtado, Adabriand, Nazareno Andrade, Nigini Oliveira, and Francisco Brasileiro (2013). "Contributor Profiles, Their Dynamics, and Their Importance in Five Q&a Sites." In: CSCW'13. ISBN: 978-1-4503-1331-5. DOI: 10.1145/2441776.2441916 (cit. on p. 13).

Gambäck, Björn and Utpal Kumar Sikdar (Aug. 2017). "Using Convolutional Neural Networks to Classify Hate-Speech." In: *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, pp. 85–90. DOI: 10.18653/v1/W17-3013 (cit. on p. 19).

Gilbert, Eric (2013a). "Widespread Underprovision on Reddit." In: *Conference on Computer Supported Cooperative Work*. CSCW '13. New York, NY, USA: ACM (cit. on p. 46).

– (2013b). "Widespread underprovision on Reddit." In: *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, pp. 803–808 (cit. on pp. 2, 18).

Grover, Aditya and Jure Leskovec (2016). "node2vec: Scalable Feature Learning for Networks." In: *International Conference on Knowledge Discovery and Data Mining* (cit. on pp. 20, 48).

Hamilton, Will, Zhitao Ying, and Jure Leskovec (2017). "Inductive representation learning on large graphs." In: *Advances in Neural Information Processing Systems* (cit. on pp. 20, 26, 39, 54).

Hammond, David K, Pierre Vandergheynst, and Rémi Gribonval (2011). "Wavelets on graphs via spectral graph theory." In: *Applied and Computational Harmonic Analysis* 30.2, pp. 129–150 (cit. on p. 34).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification." In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. USA: IEEE Computer Society, pp. 1026–1034. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.123 (cit. on p. 1).

He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua (2017). "Neural Collaborative Filtering." In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, pp. 173–182. ISBN: 978-1-4503-4913-0. DOI: 10.1145/3038912.3052569 (cit. on p. 14).

Herbrich, Ralf, Tom Minka, and Thore Graepel (2006). “TrueSkill™: A Bayesian Skill Rating System.” In: *International Conference on Neural Information Processing Systems* (cit. on p. 31).

Jamali, Mohsen and Martin Ester (2010). “A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks.” In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys ’10. Barcelona, Spain: ACM, pp. 135–142. ISBN: 978-1-60558-906-0. DOI: 10.1145/1864708.1864736 (cit. on p. 15).

Jannach, Dietmar and Malte Ludewig (2017). “When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation.” In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys ’17. Como, Italy: ACM, pp. 306–310. ISBN: 978-1-4503-4652-8. DOI: 10.1145/3109859.3109872 (cit. on pp. 8, 15).

Jenders, Maximilian, Ralf Krestel, and Felix Naumann (2016). “Which Answer is Best?: Predicting Accepted Answers in MOOC Forums.” In: *International Conference on World Wide Web* (cit. on pp. 17, 25, 27, 30, 47, 50, 52, 59, 60).

Jiang, Lu, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann (2015). “Self-paced curriculum learning.” In: *Twenty-Ninth AAAI Conference on Artificial Intelligence* (cit. on p. 1).

Kahn, Shulamit (1993). “Gender Differences in Academic Career Paths of Economists.” In: *The American Economic Review* 83.2, pp. 52–56. ISSN: 00028282 (cit. on p. 16).

Kang, Wang-Cheng and Julian McAuley (2018). “Self-Attentive Sequential Recommendation.” In: *ICDM*. IEEE Computer Society, pp. 197–206 (cit. on pp. 1, 14, 15).

Kingma, Diederik P. and Jimmy Ba (2014). “Adam: A Method for Stochastic Optimization.” In: *CoRR* abs/1412.6980. arXiv: 1412.6980 (cit. on p. 49).

Kipf, Thomas N. and Max Welling (2016a). “Semi-Supervised Classification with Graph Convolutional Networks.” In: *CoRR* abs/1609.02907. arXiv: 1609.02907 (cit. on pp. 8, 20, 26, 27, 34, 38, 48).

– (2016b). “Semi-Supervised Classification with Graph Convolutional Networks.” In: *CoRR* abs/1609.02907. arXiv: 1609.02907 (cit. on p. 20).

Kleinberg, Jon, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan (2018). “Human decisions and machine predictions.” In: *The quarterly journal of economics* 133.1, pp. 237–293 (cit. on p. 1).

Knab, Bernhard, Alexander Schliep, Barthel Steckemetz, and Bernd Wichern (2003). "Model-Based Clustering With Hidden Markov Models and its Application to Financial Time-Series Data." In: *Between Data Science and Applied Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization*. Ed. by Martin Schader, Wolfgang Gaul, and Maurizio Vichi. Springer Berlin Heidelberg. ISBN: 978-3-642-18991-3. DOI: 10.1007/978-3-642-18991-3_64 (cit. on p. 13).

Knuth, Donald E. (1974). "Computer Programming as an Art." In: *Communications of the ACM* 17.12, pp. 667–673 (cit. on p. ix).

La Thangue, Nicholas B and David J Kerr (2011). "Predictive biomarkers: a paradigm shift towards personalized cancer medicine." In: *Nature reviews Clinical oncology* 8.10, pp. 587–596 (cit. on p. 1).

Lakkaraju, Himabindu, Ece Kamar, Rich Caruana, and Jure Leskovec (2017). "Interpretable & Explorable Approximations of Black Box Models." In: *arXiv preprint arXiv:1707.01154* (cit. on p. 2).

Lakkaraju, Himabindu and Cynthia Rudin (2017). "Learning cost-effective and interpretable treatment regimes." In: *Artificial Intelligence and Statistics*, pp. 166–175 (cit. on p. 2).

Li, Qi, Fenglong Ma, Jing Gao, Lu Su, and Christopher J Quinn (2016). "Crowdsourcing high quality labels with a tight budget." In: *Proceedings of the ninth acm international conference on web search and data mining*. ACM, pp. 237–246 (cit. on p. 18).

Li, Yaliang, Nan Du, Chaochun Liu, Yusheng Xie, Wei Fan, Qi Li, Jing Gao, and Huan Sun (2017). "Reliable Medical Diagnosis from Crowdsourcing: Discover Trustworthy Answers from Non-Experts." In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, pp. 253–261 (cit. on p. 18).

Li, Yaliang, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han (2016). "A survey on truth discovery." In: *ACM Sigkdd Explorations Newsletter* 17.2, pp. 1–16 (cit. on p. 18).

Li, Yaliang, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han (2015). "On the discovery of evolving truth." In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 675–684 (cit. on p. 18).

Liu, Yong, Jorge Goncalves, Denzil Ferreira, Bei Xiao, Simo Hosio, and Vassilis Kostakos (2014). "CHI 1994-2013: mapping two decades of intellectual progress through co-word analysis." In: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, pp. 3553–3562 (cit. on p. 16).

Lu, Jiajun, Theerasit Issaranon, and David A Forsyth (2017). "SafetyNet: Detecting and Rejecting Adversarial Examples Robustly." In: *ICCV*, pp. 446–454 (cit. on p. 60).

Ma, Fenglong, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han (2015). "Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation." In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 745–754 (cit. on p. 18).

Ma, Hao, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King (2011). "Recommender Systems with Social Regularization." In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM '11. Hong Kong, China: ACM, pp. 287–296. ISBN: 978-1-4503-0493-1. DOI: 10.1145/1935826.1935877 (cit. on p. 15).

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE." In: *Journal of Machine Learning Research* (cit. on pp. 49, 51).

Maia, Marcelo, Jussara Almeida, and Virgilio Almeida (2008). "Identifying User Behavior in Online Social Networks." In: SocialNets'08. ISBN: 978-1-60558-124-8. DOI: 10.1145/1435497.1435498 (cit. on p. 13).

Mamykina, Lena, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann (2011). "Design Lessons from the Fastest Q&a Site in the West." In: CHI'11. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979366 (cit. on p. 13).

Mihaylova, Tsvetomila, Gerogi Karadzhov, Pepa Atanasova, Ramy Baly, Mitra Mohtarami, and Preslav Nakov (2019). "SemEval-2019 task 8: Fact Checking in Community question answering." In: *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pp. 856–865 (cit. on p. 18).

Mihaylova, Tsvetomila, Preslav Nakov, Lluís Marquez, Alberto Barron-Cedeno, Mitra Mohtarami, Georgi Karadzhov, and James Glass (2018). "Fact checking in community forums." In: *Thirty-Second AAAI Conference on Artificial Intelligence* (cit. on p. 17).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality." In: *Advances in neural information processing systems*, pp. 3111–3119 (cit. on p. 19).

Mishra, Pushkar, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova (Aug. 2018). "Author Profiling for Abuse Detection." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1088–1098 (cit. on p. 20).

Mishra, Pushkar, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova (2019). "Abusive Language Detection with Graph Convolutional Networks." 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)*, pp. 2145–2150 (cit. on p. 20).

Mukherjee, Tathagata, Biswas Parajuli, Piyush Kumar, and Eduardo Pasiliao (2016). "TruthCore: Non-parametric estimation of truth from a collection of authoritative sources." In: *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, pp. 976–983 (cit. on p. 18).

Nakov, Preslav, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor (2017). "SemEval-2017 task 3: Community question answering." In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 27–48 (cit. on p. 18).

Narang, Kanika, Chaoqi Yang, Adit Krishnan, Junting Wang, Hari Sundaram, and Carolyn Sutter (2019). "An Induced Multi-Relational Framework for Answer Selection in Community Question Answer Platforms." In: *arXiv preprint arXiv:1911.06957* (cit. on p. 25).

Nobata, Chikashi, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang (2016). "Abusive Language Detection in Online User Content." In: *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, pp. 145–153. ISBN: 9781450341431. DOI: 10.1145/2872427.2883062 (cit. on p. 19).

Oh, Hyo-Jung, Yeo-Chan Yoon, and Hyun Ki Kim (2013). "Finding more trustworthy answers: Various trustworthiness factors in question answering." In: *Journal of Information Science* 39.4, pp. 509–522 (cit. on p. 17).

Park, Ji Ho and Pascale Fung (2017). "One-step and Two-step Classification for Abusive Language Detection on Twitter." In: *Proceedings of the First Workshop on Abusive Language Online*, pp. 41–45 (cit. on p. 19).

Pavlopoulos, John, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos (2017). "Improved Abusive Comment Moderation with User Embeddings." In: *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pp. 51–55 (cit. on p. 20).

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "GloVe: Global Vectors for Word Representation." In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (cit. on p. 19).

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena (2014). "DeepWalk: online learning of social representations." In: *International Conference on Knowledge Discovery and Data Mining* (cit. on pp. 20, 48).

Qian, Jing, Mai ElSherief, Elizabeth Belding, and William Yang Wang (2018). "Leveraging Intra-User and Inter-User Representation Learning for Automated Hate Speech Detection." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 118–123 (cit. on p. 20).

Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme (2009). "BPR: Bayesian Personalized Ranking from Implicit Feedback." In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. UAI '09. Montreal, Quebec, Canada: AUAI Press, pp. 452–461. ISBN: 978-0-9749039-5-8 (cit. on p. 14).

Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme (2010). "Factorizing Personalized Markov Chains for Next-basket Recommendation." In: *Proceedings of the 19th International Conference on World Wide Web*. WWW '10. Raleigh, North Carolina, USA: ACM, pp. 811–820. ISBN: 978-1-60558-799-8. DOI: 10.1145/1772690.1772773 (cit. on pp. 14, 15).

Safavi, Tara, Maryam Davoodi, and Danai Koutra (2018). "Career Transitions and Trajectories: A Case Study in Computing." In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining*. KDD '18. ACM, pp. 675–684. ISBN: 978-1-4503-5552-0. DOI: 10.1145/3219819.3219863 (cit. on p. 16).

Sajjadi, Mehdi, Mehran Javanmardi, and Tolga Tasdizen (2016). "Regularization with Stochastic Transformations and Perturbations for Deep Semi-supervised Learning." In: *International Conference on Neural Information Processing Systems*. NIPS'16. ISBN: 978-1-5108-3881-9 (cit. on p. 41).

Sankar, Aravind, Adit Krishnan, Zongjian He, and Carl Yang (2019). "Rase: Relationship aware social embedding." In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8 (cit. on p. 20).

Sankar, Aravind, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang (2018). "Dynamic Graph Representation Learning via Self-Attention Networks." In: *arXiv preprint arXiv:1812.09430* (cit. on p. 20).

Santos, Tiago, Simon Walk, Roman Kern, Markus Strohmaier, and Denis Helic (Feb. 2019). "Activity Archetypes in Question-and-Answer (Q&#x38;A) Websites&#x02014;A Study of 50 Stack Exchange Instances." In: *Trans. Soc. Comput.* 2.1, 4:1–4:23. ISSN: 2469-7818 (cit. on p. 13).

Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl (2001). "Item-based Collaborative Filtering Recommendation Algorithms." In: *Proceedings of the 10th International Conference on World Wide Web*. WWW '01. Hong Kong, Hong Kong: ACM, pp. 285–295. ISBN: 1-58113-348-0. DOI: 10.1145/371920.372071 (cit. on p. 14).

Schlichtkrull, Michael, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling (2018). "Modeling relational data with graph convolutional networks." In: *European Semantic Web Conference*. Springer (cit. on pp. 20, 26, 38, 39, 47, 48, 50, 52, 54).

Schwenk, Holger and Yoshua Bengio (2000). "Boosting Neural Networks." In: *Neural Computation* 12.8, pp. 1869–1887 (cit. on p. 40).

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press (cit. on p. 61).

Silva, Leandro, Mainack Mondal, Denzil Correa, FabrıHEREcio Benevenuto, and Ingmar Weber (2016). "Analyzing the targets of hate in online social media." In: *Tenth International AAAI Conference on Web and Social Media* (cit. on p. 19).

Silver, David et al. (Jan. 2016). "Mastering the Game of Go with Deep Neural Networks and Tree Search." In: *Nature* 529.7587, pp. 484–489. DOI: 10.1038/nature16961 (cit. on p. 1).

Smyth, Padhraic (1997). "Clustering Sequences with Hidden Markov Models." In: *Advances in Neural Information Processing Systems*. MIT Press, pp. 648–654 (cit. on p. 13).

Song, Weiping, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang (2019). "Session-Based Social Recommendation via Dynamic Graph Attention Networks." In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. WSDM '19. Melbourne VIC, Australia: ACM, pp. 555–563. ISBN: 978-1-4503-5940-5. DOI: 10.1145/3289600.3290989 (cit. on p. 15).

Sukhbaatar, Sainbayar, Arthur Szlam, Jason Weston, and Rob Fergus (2015). "End-To-End Memory Networks." In: *Advances in Neural Information Processing Systems* (cit. on pp. 18, 26).

Sun, Peijie, Le Wu, and Meng Wang (2018). "Attentive Recurrent Social Recommendation." In: *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: ACM, pp. 185–194. ISBN: 978-1-4503-5657-2. DOI: 10.1145/3209978.3210023 (cit. on pp. 14, 15).

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going Deeper with Convolutions." In: *Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 1, 2).

Tan, Ming, Bing Xiang, and Bowen Zhou (2015). "LSTM-based Deep Learning Models for non-factoid answer selection." In: *CoRR* abs/1511.04108 (cit. on pp. 57–60).

Tang, Jian, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei (2015). "LINE: Large-scale Information Network Embedding." In: *International Conference on World Wide Web, WWW* (cit. on pp. 20, 48).

Tang, Jiaxi and Ke Wang (2018). "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding." In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. WSDM '18. Marina Del Rey, CA, USA: ACM, pp. 565–573. ISBN: 978-1-4503-5581-0. DOI: 10.1145/3159652.3159656 (cit. on p. 14).

Tang, Jie, Jimeng Sun, Chi Wang, and Zi Yang (2009). "Social Influence Analysis in Large-scale Networks." In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: ACM, pp. 807–816. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557108 (cit. on pp. 5, 15).

Tian, Qiongjie and Baoxin Li (2016). "Weakly hierarchical lasso based learning to rank in best answer prediction." In: *International Conference on Advances in Social Networks Analysis and Mining, ASONAM* (cit. on pp. 25, 27, 30).

Tian, Qiongjie, Peng Zhang, and Baoxin Li (2013). "Towards Predicting the Best Answers in Community-based Question-Answering Services." In: *International Conference on Weblogs and Social Media, ICWSM* (cit. on pp. 17, 25, 27, 30, 44, 47, 50, 52).

Velickovic, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio (2017). "Graph attention networks." In: *arXiv:1710.10903* (cit. on p. 26).

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio (2018). "Graph Attention Networks." In: *International Conference on Learning Representations* (cit. on p. 21).

Vydiswaran, VG, ChengXiang Zhai, and Dan Roth (2011). "Content-driven trust propagation framework." In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 974–982 (cit. on p. 18).

Wagner, Claudia, Eduardo Graells-Garrido, David Garcia, and Filippo Menczer (2016). "Women through the glass ceiling: gender asymmetries in Wikipedia." In: *EPJ Data Science* 5.1, p. 5 (cit. on p. 3).

Wan, Mengting, Xiangyu Chen, Lance Kaplan, Jiawei Han, Jing Gao, and Bo Zhao (2016). "From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1885–1894 (cit. on p. 18).

Wang, Dashun, Chaoming Song, and Albert-László Barabási (2013). "Quantifying long-term scientific impact." In: *Science* 342.6154, pp. 127–132 (cit. on p. 16).

Wang, Di and Eric Nyberg (2015). "A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering." In: *ACL* (cit. on pp. 18, 26).

Wang, Menghan, Xiaolin Zheng, Yang Yang, and Kun Zhang (2017). *Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation*. arXiv: 1711.11458 [cs.IR] (cit. on p. 15).

Wang, Xin-Jing, Xudong Tu, Dan Feng, and Lei Zhang (2009). "Ranking Community Answers by Modeling Question-answer Relationships via Analogical Reasoning." In: *SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. ACM (cit. on p. 49).

Wang, Xin, Wei Lu, Martin Ester, Can Wang, and Chun Chen (2016). "Social Recommendation with Strong and Weak Ties." In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: ACM, pp. 5–14. ISBN: 978-1-4503-4073-1. DOI: 10.1145/2983323.2983701 (cit. on p. 15).

Ward, Melanie (2001). "The gender salary gap in British academia." In: *Applied Economics* 33.13, pp. 1669–1681. DOI: 10.1080/00036840010014445 (cit. on p. 16).

Warner, William and Julia Hirschberg (2012). "Detecting hate speech on the world wide web." In: *Proceedings of the second workshop on language in social media*. Association for Computational Linguistics, pp. 19–26 (cit. on p. 19).

Waseem, Zeerak and Dirk Hovy (June 2016). "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter." In: *Proceedings of the NAACL Student Research Workshop*. San Diego, California: Association for Computational Linguistics, pp. 88–93. DOI: 10.18653/v1/N16-2013 (cit. on p. 19).

Way, Samuel F., Daniel B. Larremore, and Aaron Clauset (2016). "Gender, Productivity, and Prestige in Computer Science Faculty Hiring Networks." In: *Proceedings of the 25th International Conference on World Wide Web*. ISBN: 978-1-4503-4143-1. DOI: 10.1145/2872427.2883073 (cit. on p. 16).

Way, Samuel F., Allison C. Morgan, Aaron Clauset, and Daniel B. Larremore (2017). "The misleading narrative of the canonical faculty productivity trajectory." In: *Proceedings of the National Academy of Sciences* 114.44, E9216–E9223. ISSN: 0027-8424. DOI: 10.1073/pnas.1702121114 (cit. on p. 16).

Wen, Jiahui, Jingwei Ma, Yiliu Feng, and Mingyang Zhong (2018). "Hybrid Attentive Answer Selection in CQA With Deep Users Modelling." In: *Thirty-Second AAAI Conference on Artificial Intelligence* (cit. on p. 17).

Wong, Felix, Chee Tan, Soumya Sen, and Mung Chiang (Aug. 2016). "Quantifying Political Leaning from Tweets, Retweets, and Retweeters." In: *IEEE Transactions on Knowledge and Data Engineering* 28, pp. 1–1. DOI: 10.1109/TKDE.2016.2553667 (cit. on p. 4).

Wu, Chao-Yuan, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing (2017). "Recurrent Recommender Networks." In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. WSDM '17. Cambridge, United Kingdom: ACM, pp. 495–503. ISBN: 978-1-4503-4675-7. DOI: 10.1145/3018661.3018689 (cit. on p. 15).

Wu, Le, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang (2019). "A Neural Influence Diffusion Model for Social Recommendation." In: *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR'19. Paris, France: ACM, pp. 235–244. ISBN: 978-1-4503-6172-9. DOI: 10.1145/3331184.3331214 (cit. on pp. 8, 21).

Wu, Lingfei, Jacopo A. Baggio, and Marco A. Janssen (2016). "The Role of Diverse Strategies in Sustainable Knowledge Production." In: *PLoS ONE* (cit. on pp. 30, 31).

Wu, Wei, Houfeng Wang, and Xu Sun (2018). "Question Condensing Networks for Answer Selection in Community Question Answering." In: *Association for Computational Linguistics* (cit. on pp. 18, 26).

Wu, Yao, Christopher DuBois, Alice X. Zheng, and Martin Ester (2016). "Collaborative Denoising Auto-Encoders for Top-N Recommender Systems." In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. WSDM '16. San Francisco, California, USA: ACM, pp. 153–162. ISBN: 978-1-4503-3716-8. DOI: 10.1145/2835776.2835837 (cit. on p. 14).

Wulczyn, Ellery, Nithum Thain, and Lucas Dixon (2017). "Ex machina: Personal attacks seen at scale." In: *Proceedings of the 26th International Conference on World Wide Web*, pp. 1391–1399 (cit. on p. 19).

Yager, Ronald R (2000). "Targeted e-commerce marketing using fuzzy intelligent agents." In: *IEEE Intelligent Systems and their Applications* 15.6, pp. 42–45 (cit. on p. 1).

Yang, Jaewon, Julian McAuley, Jure Leskovec, Paea LePendu, and Nigam Shah (2014). "Finding Progression Stages in Time-evolving Event Sequences." In: *Proceedings of the 23rd International Conference on World Wide Web*. WWW'14. ACM, pp. 783–794. ISBN: 978-1-4503-2744-2. DOI: 10.1145/2566486.2568044 (cit. on p. 13).

Yang, Liu, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen (2013). “Cqarank: jointly model topics and expertise in community question answering.” In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, pp. 99–108 (cit. on p. 17).

Yang, Zhilin, William W. Cohen, and Ruslan Salakhutdinov (2016). “Revisiting Semi-Supervised Learning with Graph Embeddings.” In: *International Conference on Machine Learning, ICML*, pp. 40–48 (cit. on pp. 20, 48).

Ying, Rex, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec (2018). “Graph Convolutional Neural Networks for Web-Scale Recommender Systems.” In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD ’18*. DOI: 10.1145/3219819.3219890 (cit. on p. 20).

Yu, Bing, Haoteng Yin, and Zhanxing Zhu (2017). “Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting.” In: *CoRR* abs/1709.04875 (cit. on pp. 39, 54).

Zhang, Hengtong, Yaliang Li, Fenglong Ma, Jing Gao, and Lu Su (2018). “TextTruth: An Unsupervised Approach to Discover Trustworthy Information from Multi-Sourced Text Data.” In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, pp. 2729–2737 (cit. on pp. 6, 18).

Zhang, Xiaodong, Sujian Li, Lei Sha, and Houfeng Wang (2017). “Attentive Interactive Neural Networks for Answer Selection in Community Question Answering.” In: *AAAI Conference on Artificial Intelligence* (cit. on pp. 18, 26).

Zhang, Ziqi, David Robinson, and Jonathan Tepper (2018). “Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network.” In: *The Semantic Web*. Ed. by Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam. Cham: Springer International Publishing, pp. 745–760. ISBN: 978-3-319-93417-4 (cit. on p. 19).

Zhao, Tong, Julian McAuley, and Irwin King (2014). “Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering.” In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. CIKM ’14. Shanghai, China: ACM, pp. 261–270. ISBN: 978-1-4503-2598-1. DOI: 10.1145/2661829.2661998 (cit. on pp. 8, 15).

Zheng, Yudian, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng (2017). “Truth inference in crowdsourcing: is the problem solved?” In: *Proceedings of the VLDB Endowment* 10.5, pp. 541–552 (cit. on p. 18).

Zhou, Xiao, Desislava Hristova, Anastasios Noulas, and Cecilia Mascolo (2018). *Evaluating the impact of the 2012 Olympic Games policy on the regeneration of East London using spatio-temporal big data*. arXiv: `1807.01925 [cs.SI]` (cit. on p. 3).

Zhuang, Chenyi and Qiang Ma (2018). "Dual Graph Convolutional Networks for Graph-Based Semi-Supervised Classification." In: *World Wide Web Conference* (cit. on pp. 20, 26, 27, 39, 41, 47, 50, 52).

DECLARATION

Put your declaration here.

Urbana, Illinois , May 2020

Kanika Narang

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Thank you very much for your feedback and contribution.