Learning Representations of Social Media Users

by

Adrian Benton

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Doctor of Philosophy

Baltimore, Maryland

October, 2018

Abstract

Social media users routinely interact by posting text updates, sharing images and videos, and establishing connections with other users through friending. User representations are routinely used in recommendation systems by platform developers, targeted advertisements by marketers, and by public policy researchers to gauge public opinion across demographic groups. Computer scientists consider the problem of inferring user representations more abstractly; how does one extract a stable user representation – effective for many downstream tasks – from a medium as noisy and complicated as social media?

The quality of a user representation is ultimately task-dependent (e.g. does it improve classifier performance, make more accurate recommendations in a recommendation system) but there are also proxies that are less sensitive to the specific task. Is the representation predictive of latent properties such as a person's demographic features, socio-economic class, or mental health state? Is it predictive of the user's future behavior?

In this thesis, we begin by showing how user representations can be learned from multiple types of user behavior on social media. We apply several extensions of generalized canonical correlation analysis to learn these representations and evaluate them at three tasks: predicting future hashtag mentions, friending behavior, and demographic features. We then show how user features can be employed as distant

supervision to improve topic model fit. We extend a standard supervised topic model, Dirichlet Multinomial Regression (DMR), to make better use of high-dimensional supervision. Finally, we show how user features can be integrated into and improve existing classifiers in the multitask learning framework. We treat user representations – ground truth gender and mental health features – as auxiliary tasks to improve mental health state prediction. We also use distributed user representations learned in the first chapter to improve tweet-level stance classifiers, showing that distant user information can inform classification tasks at the granularity of a single message.

Committee:

Mark Dredze

Raman Arora

David Yarowsky

Dirk Hovy

Acknowledgments

I owe the completion of this document, and the years of work poured into it to the mentorship, collaboration, and friendship of many people. First, I am grateful to my committee for reading this document and giving me feedback after my oral exam.

Raman Arora mentored me for the multiview representation learning in this thesis. I value his patient explanations of basic linear algebra concepts and for sharing his interpretations of multiview representation learning techniques. Dirk Hovy showed me how good research should be presented, both as publications and talks. I appreciate his faith in my coding ability, his camaraderie, and his devotion to freshly-baked bread. David Yarowsky convinced me to come to Hopkins with a hard pitch for JHU during a meal at the Helmand. The pitch was effective.

Despite being my first and only Ph.D. advisor, I can confidently say that Mark Dredze is an excellent advisor. He is inexplicably optimistic when presented with lukewarm results and has always given me the freedom to pursue questions that interest me. I value the time he gave me to implement and understand the models I worked with, and the opportunity to help manage other student projects. I am grateful to him not only because he is my advisor, but also because he is a pretty good one. I hope that one day Mark will have a bubble soccer-amenable lab.

The CLSP is one of the strongest NLP groups in the world and the quality of the other students and faculty continually impressed me; it was an honor to be accepted as a student here. Adam Teichert and Michael Paul were responsible for mentoring me on several projects and gave me a crash course in machine learning debugging. Michael Paul, in particular, taught me the dark art of repairing a faulty Gibbs sampler and to rejoice when a bug is spotted. I also worked closely with Huda Khayrallah while working on applying deep generalized CCA. I appreciate our conversations and her witty reflections on grad school life. Her resourcefulness was also critical in salvaging my thesis defense.

My decision to pursue a Ph.D. was encouraged by many researchers at the University of Pennsylvania including Lyle Ungar and John H. Holmes. I am particularly grateful to Delphine Dahan for trusting a lowly undergraduate with a handful of computer science classes under his belt to join her lab, and Shawndra Hill with whom I had the closest working relationship while at Penn. Shawndra is an example of what true dedication to research looks like, always insightful and indefatigable. She continually drove me to expect more of myself.

I owe my persistence in the Ph.D. program in large part to the 2016 JSALT Social Media/Mental Health summer workshop group: Kristy, Andy, Glen, Meg, Jeff, Fatemeh, Leo, and Bu Sun. Joining this group was the best academic decision I made in the past five years. Not only was this a productive summer, it was also the most enjoyable – a refreshing blend of research and socializing. I am not sure how I would have gotten past the third-year doldrums without them.

Leo Razoumov mentored me while interning at Amazon Research and gave me the freedom to craft my own project with few constraints. He was an extremely

knowledgeable sounding board in all things mathematically rigorous and career-related. He has been a kind and supportive friend and I hope to see more of him in New York.

Many non-research relationships sustained me as well. My chess team has been a fundamental constant in my life. My parents have always supported my education and pushed me to always do my best. The O.S. & Spike Wright foundation sent pounds of trail mix and stacks of dog photos in support of thesis writing. The foundation's support was an essential component to thesis completion.

Most of all, I owe this whole grad school stint to Kika's patience, emotional support, and ultimately gentle nudging towards graduation. Thank you for sticking around this long. I love you and I hope you decide to stick around a while longer.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Social media platforms offer researchers and data scientists a massive source of user-generated data including not only what users say, but who they are friends with, their self-reported descriptions, and which posts they like. Social media data is valuable to two major groups of stakeholders: **Technologists** and social science **Researchers**. **Technologists** are focused on engineering and are concerned with either maintaining and augmenting the social media platforms themselves, or building tools that perform well on social media data. Social science **researchers** treat social media data as a lens on society, an imperfect version of how humans communicate with each other naturally. They use social media data to answer deep questions about people and the world at large, and only care about building strong tools insofar as these tools can help judge hypotheses. Each of these groups has a different set of tasks to complete and questions they want to answer.

Technologists Maintainers of social media platforms routinely use user-generated data to improve their products. These include improving the platform's friend recommendation system (Hannon, Bennett, and Smyth, 2010; Kywe, Lim, and Zhu, 2012; Konstas, Stathopoulos, and Jose, 2009) and content recommendation or feed optimization (Kramer, Guillory, and Hancock, 2014; Chen et al., 2012; Yan, Lapata, and Li, 2012; Guy et al., 2010). These features are tuned to retain users and increase the "addictiveness" of the platform. Advertising revenue is the foundation of many social media platforms' business models. Platforms such as Facebook specifically attract advertisers by using user data to better predict advertisement click-through rate.

Natural language processing (NLP) can be conceptually decomposed into an array of subtasks around automatically extracting information from human-generated text. Practitioners build tools to address each of these subtasks: part-of-speech taggers, syntactic parsers, sentiment analyzers, semantic parsers, etc. These tools were traditionally trained to perform well on standard text such as newswire, and extending them to perform well on social media posts is an active area of research (Gimpel et al., 2011; Rosenthal, Farra, and Nakov, 2017; Strauss et al., 2016; Daiber and Goot, 2016).

Researchers Social media data can also be used to test theories of how information flows through social networks (Wu et al., 2011), how these networks are structured (Martin et al., 2016), and how to identify and quantify social influence (Bakshy et al., 2011). Hypotheses which were theoretically motivated, or were empirically validated by painstakingly compiling word-of-mouth data can be tested at scale in observational studies on social media (Jansen et al., 2009). The persuasiveness of these observational studies hinges on arguing that the online behavior is evidence of a causal relationship,

and this causal argument often relies on controlling for potential confounds in social media data (Tan, Lee, and Pang, 2014).

Showing that social media data merely has predictive power for real-world happenings may also be sufficient when building predictive models. Predictive models of real-world trends such as disease incidence, stock market prices, and sentiment on public policy issues based on messages people post to Twitter can be used as surrogates for more traditional surveys (Tumasjan et al., 2010; Paul and Dredze, 2011; O'Connor et al., 2010a; Bollen, Mao, and Zeng, 2011).

In general, user-generated social media data is attractive to both groups for the following properties:

1. Social media data can be used as a proxy for actual human interactions. This is most relevant to Researchers but also to Technologists who would like to extend their systems to noisier, more naturally-produced language than news articles.

2. The data are also multi-modal and offer several views of these interactions. Take for example users' friending and messaging behavior against images or videos they post. Multiple views of user behavior can be used to build stronger tools, but can also suggest different hypotheses to test.

3. Social media platforms have a vast user base which has engorged platform servers with text and other interaction data. As-of the end of 2017, Facebook reported over 2 billion registered users, with over 17 billion video and 94 billion text chats initiated that year through their messaging feature[1]. Models trained on many examples can generalize better to out-of-sample data, which is important for both Technologists and Researchers.

[1] `https://newsroom.fb.com/news/2017/12/messengers-2017-year-in-review/`

Figure 1.1: Message on Twitter from the 45th President of the United States.

4. Social media data is produced and effectively updated in realtime. Consequently, models can be frequently updated to stay fresh and relevant. Hypotheses can also be tested as time progresses to validate that past findings hold in the present.

Some benefits such as quantity and rate of updating are shared with other online data sources such as server and web search logs. However, these other data sources do not capture natural human interactions.

1.1.1 Problems Associated with Social Media Data

Social media data can be a source for building robust NLP tools, predictive models of real-world trends on online activity, and testing social scientific theories. However, there are several fundamental problems that anyone who wants to build tools using these data must overcome. Figure 1.1 is an example of a message posted on Twitter (tweet) that demonstrates many of these problems[2].

Feature Sparsity NLP models often rely on token n-gram features to make predictions. Longer document lengths allow these models to generalize well. On social media, messages with only a handful of tokens are common, leading to very sparse feature vectors. Additional sparsity arises from the fact that conversations on social

[2] `https://www.cnn.com/2017/05/31/politics/covfefe-trump-coverage/index.html`

Figure 1.2: Following tweet with the mysterious "covfefe".

media span many domains, even when one is restricting to messages made in English, for instance. This is further exacerbated by frequent typos, butter fingers, and intentionally alternate spellings. The vocabulary size is larger for social media messages than restricted domains such as Wall Street Journal articles.

Context Most importantly, the **context** of this tweet is absent from the tweet itself. "*Despite the negative press covfefe*" is not a complete English sentence. However, knowing that the user posting this tweet is the current President of the United States of America who has a confrontational history with the press, one can infer that "covfefe" was meant to be the word "coverage" and that this was meant to be followed by some self-aggrandizing statement. Context can also include previous activity within the social media platform. In this case, context can be previous messages within a discussion, or other messages posted with similar content. Figure 1.2 shows another message where humor is dependent on awareness of the original presidential *covfefe*. Lack of context is a fundamental problem in NLP, since natural language regularly refers to events and entities in the real world, but it is exacerbated on social media primarily because of short message length.

1.1.2 User Features to Alleviate Problems with Social Media

Knowing the context around a social media message is key to understanding its meaning. Author demographic features, such as age, gender, and socio-economic class are an old and critical piece of this context. Demographic features are traditionally treated as categorical, where users that fall in the same demographic category can be thought of as belonging to the same "hard cluster".

Hard User Clusters Partitioning a population by some subset of stable properties and then describing the behavior of each of these subgroups is one way to use user features to improve social media systems. This idea has been applied in several fields:

1. **Marketing**: *Market segmentation*, particularly *demographic segmentation* is a classic marketing strategy. Smith (Smith, 1956) described market segmentation in 1956, and contrasted this strategy with product differentiation, which refers to supply side heterogeneity (distinguishing one's product from the competition). Although Smith considered market segmentation abstractly: "Segmentation is based upon developments on the demand side of the market and represents a rational and more precise adjustment of product and marketing effort to consumer or user requirements", demographic quantities such as gender and age typically defined the boundaries between different market segments. This was because these boundaries aligned with dominant stereotypes (e.g. men buy lawnmowers and women buy dish soap) and these characteristics could be reliably quantified. This strategy has been transplanted to political campaigns as well – instead of hawking soap, campaigns sell a candidate or policy platform. More fine-grained targeting has also been used to better appeal to consumers.

These include using psychographic properties to define groups or identifying consumers with specific interests or habits (gardeners, bicyclists, latte-drinkers), as is offered by the Facebook advertising platform[3]. Nevertheless, partitioning the market into broad clusters based on a set of categorical indicators is the norm.

2. **Computational social science/policy**: Social scientists and public policy researchers are interested in characterizing a population by different groups for essentially the same reason as marketing scientists: policy opinions or beliefs are not homogeneously distributed throughout a population. Public policy surveys reflect this by disaggregating public opinion by different groups, often along demographic features. Exploration of the best subset of features to segment a population into homogeneous subgroups is an active area of research in public health (Boslaugh et al., 2004).

3. **NLP**: There has also been interest in using author features to improve performance at standard NLP tasks including sentiment analysis and part-of-speech tagging (Hovy, 2015). Inferring latent user features from social media has been thoroughly explored in NLP, either predicting typical demographic properties (Volkova et al., 2015a) or less typical features such as profession or interests (Beller et al., 2014).

Problems with Using Hard User Clusters in Social Media Although this is an attractive solution when demographic features are available, many social media platforms do not share user demographics. Even when demographic features are available,

[3] `https://www.facebook.com/business/products/ads`

by hand annotation for example, the labels may be influenced by the annotators' biases (e.g. a user is coded as black because of how they tweet) in hand annotation, or reporting bias when demographics are self-reported. Fitting classifiers and tuning systems on disjoint subsets of users will not work well on small training sets, harming performance compared to engineering systems on the full training set. More importantly, it is not clear a priori which user features we should be conditioning on for each task.

1.2 Proposal: User Embeddings

We instead propose learning distributed user embeddings based on a user's online behavior. A user embedding, in this context, is a vector of real numbers, that succinctly captures properties of that user. Users with similar embeddings should behave similarly on social media.

The process of learning user embeddings is inspired by work in NLP on learning word and generally text embeddings. Word embeddings are vector representations of words where closeness is able to capture semantic and syntactic relationships between words. The key component in learning word embeddings is that they are trained to be predictive of the word's context, where context is defined as the identity of surrounding words (collocations) (Mikolov et al., 2013a). Surrounding words make for good context when learning word embeddings, but there is no clear analog for "user context".

Social media user activity is arguably much richer than the appearance of a word in documents. A user can be characterized by the friends they connect to, the messages they post, or the articles and images they share. In order to incorporate multiple

behaviors, we propose using multiview representation learning techniques to learn user representations that capture multiple views of online activity simultaneously. We also learn user embeddings with standard dimensionality reduction techniques and evaluate their effectiveness at several downstream social media tasks.

Continuous Relaxations of Other Hard Clusters Using vector-valued user embeddings rather than categorical user features to improve classifiers is an old idea and has analogs in several other domains:

- Recommendation systems can be broadly categorized into those that make recommendations based on which subgroup a user belongs to (content-based filtering) vs. those that make recommendations based on preferences of similar users, where similarity is defined as "rating items similarly" (collaborative filtering). Content-based methods cluster users by provided features, such as given demographics or stated genre preferences, and make recommendations based on similarity in this feature space, although these user representations may also be distributed (e.g. a user embedding learned based on a free text description field) (Adomavicius and Tuzhilin, 2005). Collaborative filtering approaches based on factorization of a large user-item matrix are similar in spirit to our distributed user embeddings.

- In NLP, distributed word embeddings were preceded by Brown clusters, which are hierarchical cluster representations of words (Brown et al., 1992). Words that share more parent nodes in this hierarchy tend to be more similar semantically than those that do not.

1.3 Contributions

In this thesis we learn distributed social media user embeddings and evaluate how well these embeddings and traditional user features improve downstream tasks. In the process, we present machine learning models to both learn and use user features. There are two main thrusts of this thesis: methodological contributions in learning user embeddings, and evaluating these user features at improving downstream tasks.

We learn the following user embeddings:

- Principal component analysis (PCA) embeddings of the *ego* user's message text. We also consider PCA reductions of different user activity views such as the *friend*, *follower*, *mention*ed user networks, as well as reductions of the text of those groups.

- Generalized canonical correlation analysis (GCCA) derived embeddings, where user views are different types of user activity. We present two novel, orthogonal extensions of GCCA: to discriminatively weight the reconstruction error of each view, and to learn nonlinear transformations from observed to latent space.

We evaluate user features and pretrained user embeddings at improving performance at the following tasks:

- **User-level hashtag prediction**: Predicting whether or not a Twitter user will use a particular hashtag in a future tweet.

- **Friend recommendation**: Predicting whether Twitter users have established a friend relationship with each other.

- **Demographic prediction**: Predicting users' age, gender, or political affiliation.

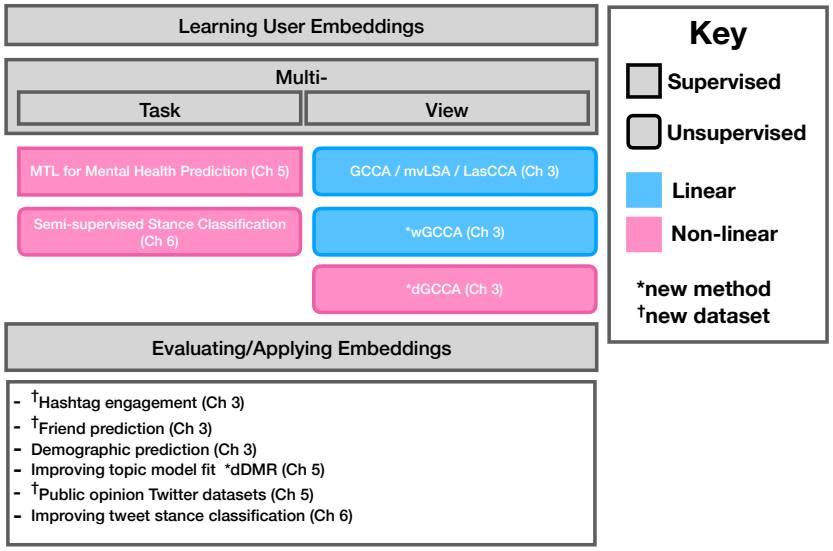


Figure 1.3: Classification scheme of methods we explore for learning user embeddings and how they are evaluated.

- **Topic model fit**: Improving the quality and fit of supervised topic models on corpora of social media posts.

- **Tweet-level stance classification**: Predicting the opinion expressed in a tweet with respect to a specific issue.

Model Implementations: A main contribution of this thesis are the publicly released implementations of models presented here. Many of these methods cannot be applied naïvely to real-world datasets – scaling them up to the number of examples and feature dimensionality present in real-world datasets, accounting for missing data, and differentially weighting the importance of views are all non-trivial challenges. Table 1.1 lists the method implementations we have made public.

Method	**Model Class**	**Features**	URL
wGCCA	multiview method	Supports missing data per example-view and weighting of views	`https://github.com/abenton/wgcca`
LasCCA	multiview method	Supports missing data per example-view	`https://github.com/abenton/MissingView-LasCCA`
dGCCA	multiview method	Supports missing data per example-view and learn neural embedding of views	`https://bitbucket.org/adrianbenton/dgcca-py3`
dDMR	topic model	Learn neural embedding of document supervision	`https://github.com/abenton/deep-dmr`

Table 1.1: Publicly-released implementations of methods presented in this thesis. The **Features** column highlights contributions of this implementation over existing implementations. The **URL** where each implementation is currently hosted is noted in the rightmost column.

Since publication, these implementations have been used by researchers to learn relationships between many different kinds of data sources such as substance abuse and social media language (Ding, Bickel, and Pan, 2017), speech and cognitive impairment features[4], as well as to learn multimodal representations of video (Tsai and Kender, 2017).

1.4 Overview

We present several methods for learning embeddings and evaluate according to many objectives. Figure 1.3 classifies the different methods we explore, evaluation tasks we

[4]The *wGCCA* implementation was shared with the *Remote Monitoring of Neurodegeneration through Speech* team at the Third Frederick Jelinek Memorial Summer Workshop (JSALT 2016). The *dGCCA* implementation was also extended and applied by the *Grounded Sequence to Sequence Transduction* team at JSALT 2018.

perform, and what function each serves in this thesis. Newly-developed models and new datasets are denoted by * and †, respectively.

Chapter 2 contains background on user features and embeddings in social media research as well as the methods we use in this thesis to learn and utilize learned user embeddings: multiview representation learning and multitask learning.

Chapter 3 describes how user embeddings can be learned by unsupervised multi-view learning techniques, and analyzes the efficacy of different views on downstream embedding performance in predicting which hashtags a Twitter user will mention, who they will friend, and their demographic features. We present an extension of generalized canonical correlation analysis (GCCA): weighted GCCA to learn user embeddings. Weighted GCCA and the demographic prediction experiments were presented in Benton, Arora, and Dredze (2016), published as a short paper in the Proceedings of the Conference of the Association for Computational Linguistics (ACL) in 2016. The experiments with deep GCCA were presented in Benton et al. (2017), an arχiv preprint. Robust LasCCA was developed and implemented during an internship at Amazon Research.

In Chapter 4, we evaluate using (distant) author-level features to better fit topic models to social media messages. We then describe a supervised topic model that can make effective use of user embeddings to better fit social media text, in lieu of explicit author features: deep Dirichlet Multinomial Regression. This model was originally presented in Benton and Dredze (2018a), published as a long paper in the Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL). The distant author feature topic model experiments were presented in Benton et al. (2016b), a long paper published in

the Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), 2016.

Chapter 5 describes work in leveraging several Twitter user mental conditions to better predict suicide risk from their social media posts. It also considers how including user features such as demographics as an auxiliary task can improve mental condition prediction. This work was published in Benton, Mitchell, and Hovy (2017), a long paper in the Proceedings of the European Chapter of the Association for Computational Linguistics (EACL), 2017.

Chapter 6 describes a final application of embeddings where we show how the embeddings learned in chapter 3 can be used to learn stronger tweet stance classification models in a multitask learning framework. Neural classifiers can be pretrained to predict generic user embedding features for a general set of users before being finetuned on a specific task. This can alternately be read as an extension of chapter 5 to treating learned user embeddings as auxiliary tasks, not categorical supervision. This work was presented at the 4th Workshop on Noisy User-Generated Text (W-NUT) (Benton and Dredze, 2018b).

Chapter 7 summarizes the contribution of each chapter and provides direction for future work.

Chapter 2

Background

This chapter begins in Section 2.1 with a discussion of existing work in applying user features to improve downstream systems, followed by sections on methods we will use to learn user representations and integrate them into existing models. Section 2.2 can be read as a primer on multiview representation learning that covers the basics of canonical correlation analysis and extensions to more than two views and nonlinear mappings. We primarily use these methods to learn user embeddings in Chapter 3. Section 2.3 finally describes the multitask learning paradigm, which is used to inject user information into trained models in Chapters 5 and 6.

2.1 Applications of User Features

User features and representations have been shown to help in a variety of downstream tasks. Here we give a selection of tasks that benefit from stronger information about the user.

2.1.1 Inferring Latent User Features

We rarely have direct access to latent user features such as gender, personality, socioeconomic class, or political affiliation. Models that can infer these traits are particularly desirable since the demographics predictions can be used as proxies for many different kinds of behaviors we may want to predict.

Demographics Most work predicting latent social media user features has been focused on predicting user demographic features (Volkova et al., 2015a). These include predicting properties such as age (Rao et al., 2010; Nguyen, Smith, and Rosé, 2011; Nguyen et al., 2013), gender (Al Zamal, Liu, and Ruths, 2012; Culotta, Ravi, and Cutler, 2016), race or ethnicity (Pennacchiotti and Popescu, 2011; Preoțiuc-Pietro and Ungar, 2018), socioeconomic category (Volkova et al., 2015a; Culotta, Ravi, and Cutler, 2016), and political affiliation (Pennacchiotti and Popescu, 2011; Cohen and Ruths, 2013; Volkova, Coppersmith, and Van Durme, 2014a). The standard approach is to train a supervised model to predict one of these properties given a corpus of annotated, observed user data. These models either treat the demographic prediction problem as regression (Nguyen, Smith, and Rosé, 2011) (e.g. predicting an author's age), or classification (Pennacchiotti and Popescu, 2011) (e.g. predicting an author's gender).

Standard features include word (Rao et al., 2010) and character n-gram features (Pennacchiotti and Popescu, 2011) of messages users post, output from NLP systems such as word stems or part of speech tags (Al Zamal, Liu, and Ruths, 2012; Nguyen et al., 2013; Preoțiuc-Pietro and Ungar, 2018), and topic and word embedding features (Pennacchiotti and Popescu, 2011; Preoțiuc-Pietro and Ungar, 2018). Dictionary

features such as the Linguistic Inquiry and Word Count (LIWC) are also popular, especially since these features have been shown to correlate with meaningful demographic and psychometric user properties (Tausczik and Pennebaker, 2010). It is also common to draw on features of the local network such as the identities of neighboring users or text that they post (Culotta, Ravi, and Cutler, 2016; Yang and Eisenstein, 2017). This is done either by aggregating information from friends or followers of the source user into a feature vector (Al Zamal, Liu, and Ruths, 2012), or by sharing predictions made on neighboring users through the social graph (Yang and Eisenstein, 2017). Work that takes the latter approach exploits *homophily*, the tendency of similar users to establish connections with others in the social graph.

Mental Properties Although more recent, there is also a community around predicting less traditional user properties such as mental health (De Choudhury et al., 2013; Coppersmith et al., 2015a; Coppersmith et al., 2016) and user personality (Schwartz et al., 2013a; Schwartz et al., 2014; Preoţiuc-Pietro et al., 2015). Similar to predicting demographics, the typical approach is to train supervised models to predict these features. A major difficulty with learning mental health classifiers is that unlike demographic features such as gender which are relatively easy to annotate, mental health is a particularly sensitive characteristic. Not only are subjects reticent in divulging this information, but care should be taken by researchers, even when mental health status is inferred (Benton, Coppersmith, and Dredze, 2017).

One clever approach is to consider public messages self-reporting having a particular condition as genuine (Harman, Coppersmith, and Dredze, 2014; Coppersmith et al., 2015b). Padrez et al. (2015) assembled a parallel corpus of electronic health records alongside Twitter and Facebook posts. However, these subjects manually

opted in from a single hospital emergency department, and therefore the number of positive examples for any single mental health condition is small.

Personality is a less sensitive target to predict, since users regularly subject themselves to online personality tests (Kosinski, Stillwell, and Graepel, 2013; Plank and Hovy, 2015). Nevertheless, knowing user personality has implications for predicting future behavior (Cadwalladr and Graham-Harrison, 2018), and it is unclear how comfortable users are with personalized inferences made about them without their consent.

2.1.2 Recommendation Systems

Recommendation systems can be grouped into two main classes based on how recommended items are ranked: *collaborative filtering* and *content-based*. Content-based systems are more strongly dependent on user profile since recommendations are based on representations of the user and the item being recommended. Collaborative filtering systems make recommendations based on prior consumption. Collaborative filtering systems have a hard time making useful recommendations early on because they rely on a history of the user's consumption. This is known as the cold-start problem (Adomavicius and Tuzhilin, 2005). Content-based systems are not as susceptible to the cold-start problem, since they can make recommendations based on extraneous user factors (e.g. a user description that is populated at enrollment). Although systems rarely fall squarely in one category or the other, this remains a useful dichotomy.

Recommendation systems on social media platforms either recommend *content* to consume or recommend *friends* to connect with (Phelan, McCarthy, and Smyth, 2009; Leskovec, Huttenlocher, and Kleinberg, 2010). Facebook and Twitter news feeds are examples of content recommendation systems operating over the space of other

users' messages. Predicting whether or not a user would click on an advertisement can also be viewed as a recommendation system, where the items that are recommended are advertisements for various products (Lohtia, Donthu, and Hershberger, 2003; Dembczynski, Kotlowski, and Weiss, 2008). In this case, clicking on an advertisement constitutes consuming the item. Note that a content-based approach, modeling the user, is critical since ad clicks are very rare events (Wang et al., 2011).

2.1.3 Social Science

Social media data provides researchers with a platform to study the effects of human relationships, social networks, on behavior. One goal of social media analytics is to replace traditional survey mechanisms (Thacker and Berkelman, 1988; Krosnick, Judd, and Wittenbrink, 2005) by monitoring messages posted on social media. Although the surveys that are simulated are most regularly seen in political polling (Tumasjan et al., 2010), they also appear in tracking disease and public health (Paul and Dredze, 2011; Culotta, 2014), and opinion related to public policy issues (O'Connor et al., 2010a; Stefanone et al., 2015; Benton et al., 2016a).

However, social media users are a biased sample relative to the general population (Ruths and Pfeffer, 2014). This presents difficulties when predicting survey responses directly from online messages. One way to account for difference in the populations is to adjust one's predictions based on demographic features of the social media population you are measuring. Inferred demographics can be used to appropriately adjust for bias on social media (Culotta, 2014).

User features are also important to control for as potential confounds when measuring influence in social networks (Hill, Provost, and Volinsky, 2006). Aral, Muchnik, and Sundararajan (2009) find that features such as a user's demographics explain most

of the tendency to adopt a mobile application in an instant messaging network. Not controlling for homophily in the network means that the effect of social influence – one user adopting the application leads their friends to adopt it – is overestimated since users with a natural propensity to adopt will share other features that also make them more likely to be linked. Global position in the social network may also be used as a substitute for latent user features (Hill et al., 2011).

2.1.4 Message-Level Prediction

User information can help systems make better predictions for single messages/documents even when not clearly related to the message-level prediction task. Work related to improving NLP systems by conditioning on user demographics is a key example. Hovy (2015) show that training separate classifiers for product reviewers of different gender and age can improve accuracy at predicting product category and rating. Johannsen, Hovy, and Søgaard (2015) show that author gender is predictive of certain types of syntactic patterns in online product reviews. This suggests that knowing features of the user writing a review could improve syntactic parsing of sentences. Similarly, Hovy and Søgaard (2015) show that author age affects the performance of already trained part-of-speech taggers, suggesting a disparity in the way younger vs. older authors use language.

Instead of relying on ground truth user features, messages can be conditioned on generic user embeddings. For example, Amir et al. (2016) finds that tweet sarcasm detection can be improved by augmenting the input features with pretrained user embeddings as a source of context.

2.2 Multiview Representation Learning

2.2.1 Motivation

The goal of multiview representation learning is to learn representations for a class of objects that capture correspondences between multiple feature sets, *views*, associated with each object. We learn these representations because we believe they will be predictive of some latent object property, a useful component in a downstream system. These feature sets often correspond to multiple modalities. Take for example the X-ray microbeam dataset, a corpus of speech utterances containing acoustic measurements paired with the position of speech articulators (Westbury, 1994). Multiview learning methods have successfully been applied to this data to learn representations predictive of what phone a person is uttering at each frame (Wang et al., 2015).

Multiview methods are applied under the assumption that each view is sufficient to predict a target of interest *given enough training data* (Kakade and Foster, 2007). However, we almost never have enough training data, so variance in our small training set will obscure the mapping from input features to target. By learning a representation of what is common between views, discarding uncorrelated noise, we ignore uninformative variance in our input features and yield better downstream performance. Single-view dimensionality reduction techniques such as principal component analysis may discard variance in the data that happens to be correlated across views, simply because it treats the input features as a single feature set.

In this thesis, we learn multiview user embeddings derived by applying variants of canonical correlation analysis (CCA), an old statistical technique for finding linear transformations of two random variables such that they are maximally correlated (Hotelling, 1936). In this section we describe the CCA problem and present solution

derivations. We also describe objectives extending CCA to learn nonlinear mappings between two views, nonlinear kernel CCA and deep CCA. We finally discuss MAX-VAR generalized CCA, an extension to maximizing correlation between more than two views. See Uurtio et al. (2017) for another CCA tutorial, a discussion about using it as an analysis tool, and interpreting the learned embeddings.

Other Multiview Techniques Although in this thesis we learn user embeddings with methods related to CCA, there is a long history of using multiview techniques to learn representations as well as classifiers. Below is a selection of related methods.

Co-training is a semi-supervised approach for training a robust classifier from few labeled examples (Blum and Mitchell, 1998). In this method, the feature set is partitioned into two views, and an independent classifier is trained independently on each view. An unlabeled dataset is then tagged by each classifier, and the unlabeled data along with the predicted labels are used to augment the other classifier's training set. This entrains each classifier to make similar predictions from different feature sets. This framework has applicability beyond learning classifiers, and has also been applied to the problem of multiview clustering (Kumar, Rai, and Daume, 2011).

Siamese networks are a class of neural models that can be applied to multiview representation learning (Bromley et al., 1994). In this framework, each view is passed through a network and network weights are trained to minimize the ℓ_2 distance between the siamese network output layers. This is similar in spirit to CCA, where as we will show, correlation between two views is maximized.

Another related class of models are multiview probabilistic generative models, where a latent variable is assumed to govern the distribution of several observed views. Topic models that infer a shared distribution over topics for multiple document views

(e.g. the body and title of a news article) are one class of models (Ahmed and Xing, 2010). CCA has a corresponding probabilistic model as well (Section 2.2.2.4).

2.2.2 Canonical Correlation Analysis

CCA is a statistical technique used to learn a linear relationship between two sets of random variables. These two sets of variables are referred to as *views*. CCA is applied when one wants to maximize correlation between views and discard independent variation as noise.

2.2.2.1 Problem Definition

Suppose we are given two data matrices $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$ where X corresponds to view 1, and Y corresponds to view 2. n is the number of examples in your data, p is the number of features in view 1, and q is the number of features in view 2.

The one-dimensional CCA problem is as follows:

$$\begin{aligned} &\max_{u \in \mathbb{R}^p, v \in \mathbb{R}^q} z_X^T z_Y \\ &\text{where } z_X = Xu; \qquad z_Y = Yv \\ &\text{subject to } \|z_X\|_2 = 1; \quad \|z_Y\|_2 = 1 \end{aligned} \tag{2.1}$$

The solutions to this problem, u and v, are points in the feature space that are mapped to points in $\mathbb{R}^n$ by the view data matrices. u and v are called the *canonical weight vectors* or *canonical weights*, and their images under X and Y, z_X and z_Y, are called the *canonical variates*. This is the one-dimensional CCA problem, as we are finding a single pair of canonical weights. It can be extended to finding more than one set of canonical weight vectors by solving for u^i and v^i that satisfy the above problem,

with the additional constraints that z_X^i and z_Y^i are orthogonal to all other z_X^j and z_Y^j. For all $i \in 1 \ldots k$, the k-dimensional CCA problem then becomes:

$$\begin{aligned} &\max_{u^i \in \mathbb{R}^p, v^i \in \mathbb{R}^q} (z_X^i)^T z_Y^i \\ \text{subject to } &\|z_X^i\|_2 = 1; \qquad \|z_Y^i\|_2 = 1 \\ &\forall j < i, \qquad (z_X^i)^T z_X^j = 0 \\ &\qquad\qquad (z_Y^i)^T z_Y^j = 0 \end{aligned} \tag{2.2}$$

Why *Correlation* Analysis? If we consider z_X and z_Y to be n draws of two scalar-valued random variables, then the empirical correlation between these variables is $\frac{1}{n-1} \frac{z_X^T z_Y}{\sqrt{z_X^T z_X}\sqrt{z_Y^T z_Y}}$.

The constraints in the CCA objective ensure that z_X and z_Y are both unit-norm, so:

$$\begin{aligned} corr(z_X, z_Y) &= \frac{1}{n-1} \frac{z_X^T z_Y}{\sqrt{z_X^T z_X}\sqrt{z_Y^T z_Y}} \\ &= \frac{1}{n-1} \frac{z_X^T z_Y}{\sqrt{1}\sqrt{1}} \\ &= \frac{z_X^T z_Y}{n-1} \end{aligned}$$

This quantity is maximized when the inner product between z_X and z_Y is maximized.

2.2.2.2 Notation and Terminology

Suppose we are given two data matrices $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$ where X corresponds to view 1, and Y corresponds to view 2. n is the number of examples in the data, p is the number of features in view 1, and q is the number of features in view 2.

To simplify the following derivations, we assume that the columns of each of these matrices are normalized such that their means are zero and have unit variance[1]. We assume, without loss of generality, that $p \leq q$.

Useful Definitions The sample *auto-covariance* matrices for views 1 and 2:

$$C_{XX} = \frac{1}{n-1} X^T X$$
$$C_{YY} = \frac{1}{n-1} Y^T Y$$

The sample *cross-covariance* matrices between views 1 and 2:

$$C_{XY} = \frac{1}{n-1} X^T Y$$
$$C_{YX} = \frac{1}{n-1} Y^T X$$

Note that $C_{XY} = C_{YX}^T$. Finally, the *joint covariance* matrix for both views:

$$\begin{pmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{pmatrix}$$

This is the covariance matrix in the single-view setting, the auto-covariance matrix for the concatenation of both views. However, it will be useful to consider each block of this matrix separately, since they correspond to the auto-covariance and cross-covariance matrices of the individual matrices.

[1]If you are given views that do not satisfy these constraints, they can be normalized simply by subtracting the mean from each column and dividing each feature value by the column standard deviation. Remember to save these data means and standard deviations used in preprocessing, so they can be applied to test data.

2.2.2.3 Solution

Below are two sketches of derivations for solving the CCA problem. The first derivation was given in Hotelling (1936), and the second was published over 50 years later in Ewerbring (1990).

Original Hotelling Derivation At a high-level, the original solution presented by Hotelling gives the solution for the first pair of canonical weights and variates. Iit boils down to the following steps:

1. Form the augmented Lagrangian of the CCA problem.
2. Take the partial derivatives of the augmented Lagrangian with respect to the unknowns.
3. Mathematically massage these equations to yield an eigenvalue problem.
4. Show that the eigenvectors solutions to this equation are one set of canonical weights, and the eigenvalues are correlations between canonical variates of the CCA problem.
5. Solve for the other set of canonical weights by substitution.

The first observation is that $\text{CORR}(z_X, z_Y)$ does not change if we scale z_X or z_Y, so let us scale u and v to ensure they are both unit-norm.

$$\begin{aligned} z_X^T z_X = u^T X^T X u &= u^T C_{XX} u = 1 \\ z_Y^T z_Y = v^T Y^T Y v &= v^T C_{YY} v = 1 \end{aligned}$$

Thus, we can rewrite the CCA problem as:

$$\begin{aligned} &\max_{u,v} \langle Xu, Yv \rangle = u^T C_{XY} v \\ \text{subject to } &u^T C_{XX} u = 1 \\ &v^T C_{YY} v = 1 \end{aligned} \tag{2.3}$$

First we use the Lagrange multipler technique to fold the constraints into the objective with Lagrange multipliers λ_X and λ_Y:

$$\mathcal{L}(u, v, \lambda_X, \lambda_Y) = u^T C_{XY} v - \lambda_X (u^T C_{XX} u - 1) - \lambda_Y (v^T C_{YY} v - 1) \tag{2.4}$$

We take the partial derivative of the righthand side with respect to u and v, and set each equal to 0 – a solution of the objective must necessarily also be a stationary point of the Lagrangian for some non-negative values λ_X and λ_Y.

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta u} &= C_{XY} v - 2\lambda_X C_{XX} u &= 0 \\ \frac{\delta \mathcal{L}}{\delta v} &= C_{YX} u - 2\lambda_Y C_{YY} v &= 0 \end{aligned} \tag{2.5}$$

Multiply each equation on the left by u^T and v^T, respectively.

$$\begin{aligned} u^T C_{XY} v - 2\lambda_X u^T C_{XX} u = 0 \\ v^T C_{YX} u - 2\lambda_Y v^T C_{YY} v = 0 \end{aligned}$$

We know that a solution to the CCA problem must satisfy the unit norm constraints constraints, so we insert these:

$$\begin{aligned} u^T C_{XY} v &= 2\lambda_X \\ v^T C_{YX} u &= 2\lambda_Y \end{aligned} \tag{2.6}$$

and since the left-hand side of the first equation is just the transpose of the second (and is a scalar), we know that the multipliers must be the same value $\lambda = \lambda_X = \lambda_Y$. Substituting for λ back into Equation 2.5 yields:

$$\begin{aligned} C_{XY} v &= 2\lambda C_{XX} u \\ \frac{C_{XX}^{-1} C_{XY} v}{2\lambda} &= u \\ C_{YX} u &= 2\lambda C_{YY} v \\ \frac{C_{YY}^{-1} C_{YX} u}{2\lambda} &= v \end{aligned} \tag{2.7}$$

Note that C_{XX} and C_{YY} are invertible because they are both symmetric – $X^T X = (X^T X)^T$ – and positive definite – $\forall w \in R^p, (Xw)^T (Xw) > 0$[2]. A final substitution of u from Equation 2.7 into Equation 2.5 yields:

$$\begin{aligned} \frac{C_{YX} C_{XX}^{-1} C_{XY} v}{2\lambda} &= 2\lambda C_{YY} v \\ (C_{YY}^{-1} C_{YX} C_{XX}^{-1} C_{XY}) v &= 4\lambda^2 v \end{aligned} \tag{2.8}$$

This is in the form of an eigenvalue problem, where v is the principal eigenvector for the left-hand side matrix and $4\lambda^2$ is its associated eigenvalue. We can use an

[2]We are assuming the auto-covariance matrices are is full-rank and $\|w\| > 0$. This is a reasonable assumption if $n >> p, q$.

eigensolver to solve for v. Once solved, we can substitute v back into Equation 2.5 and finally solve for u.

This derivation can be extended to finding k pairs of canonical weights by replacing u and v by matrices $U \in \mathbb{R}^{p \times k}$ and $V \in \mathbb{R}^{q \times k}$, where each successive column V^i is the i^{th} eigenvector of $C_{YY}^{-1} C_{YX} C_{XX}^{-1} C_{XY}$ and U is solved by substitution into Equation 2.7. In summary, assuming that C_{XX} and C_{YY} are invertible, a solution to the k-dimensional CCA problem can be found by:

$$\begin{aligned} V &= k \text{ top eigenvectors of } C_{XX}^{-1} C_{XY} C_{YY}^{-1} C_{YX} \\ U &= \frac{C_{XX}^{-1} C_{XY} V}{2\lambda} \end{aligned} \tag{2.9}$$

SVD of Joint Covariance Matrix A second derivation of the CCA solution expresses the CCA objective in terms of the data's joint covariance matrix (Ewerbring, 1990). The constraint that successive canonical variates be orthogonal to each other and unit-norm can be written as:

$$U^T C_{XX} U = I$$
$$V^T C_{YY} V = I$$

where if $U \in \mathbb{R}^{p \times k}$ and $V \in \mathbb{R}^{q \times k}$, with $k <= p$, then I is the $k \times k$ identity matrix. The objective can also be written as:

$$U^T C_{XY} V = \Lambda$$

where $\Lambda \in \mathbb{R}^{k \times k}$ is a diagonal matrix whose diagonal values, $\Lambda_{1,1} \ldots \Lambda_{k,k}$, are the canonical correlations, $(z_X^i)^T z_Y^i$. We can express both the constraints and the objective in a single equation:

$$\begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix} \begin{pmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{pmatrix} \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} = \begin{pmatrix} I & \Lambda \\ \Lambda & I \end{pmatrix} \tag{2.10}$$

If we let $\widetilde{U} = C_{XX}^{\frac{1}{2}} U$ and $\widetilde{V} = C_{YY}^{\frac{1}{2}} V$, we can rewrite Eq. 2.10 as:

$$\begin{pmatrix} \widetilde{U}^T & 0 \\ 0 & \widetilde{V}^T \end{pmatrix} \begin{pmatrix} I & C_{XX}^{-\frac{1}{2}} C_{XY} C_{YY}^{-\frac{1}{2}} \\ C_{YY}^{-\frac{1}{2}} C_{YX} C_{XX}^{-\frac{1}{2}} & I \end{pmatrix} \begin{pmatrix} \widetilde{U} & 0 \\ 0 & \widetilde{V} \end{pmatrix} = \begin{pmatrix} I & \Lambda \\ \Lambda & I \end{pmatrix}$$

From the on-diagonal blocks, we know that $\widetilde{U}$ and $\widetilde{V}$ must be orthogonal matrices, and we can manipulate the off-diagonal elements to show that:

$$\begin{aligned} \widetilde{U}^T C_{XX}^{-\frac{1}{2}} C_{XY} C_{YY}^{-\frac{1}{2}} \widetilde{V} &= \Lambda \\ C_{XX}^{-\frac{1}{2}} C_{XY} C_{YY}^{-\frac{1}{2}} &= \widetilde{U} \Lambda \widetilde{V}^T \end{aligned} \tag{2.11}$$

We can solve for $\widetilde{U}$, $\widetilde{V}$, and Λ by a rank-k truncated singular value decomposition (SVD) of the left-hand matrix, then solve for the canonical weights by $U = C_{XX}^{-\frac{1}{2}} \widetilde{U}$ and $V = C_{YY}^{-\frac{1}{2}} \widetilde{V}$.

Why these Derivations are Interesting The first derivation was originally presented in Hotelling (1936). This second derivation is worth seeing since we can learn both pairs of canonical weights using an SVD of a particularly constructed matrix. It also relates the CCA weights to the joint sample covariance matrix. One can draw

connections to other linear techniques such as principal component analysis, where an eigendecomposition of the joint covariance matrix yields the principal components:

$$\begin{pmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{pmatrix} = U^T \Lambda U \tag{2.12}$$

2.2.2.4 Probabilistic Interpretation

Bach and Jordan (2005) showed that the solution for the CCA objective is equivalent to the maximum likelihood solution of latent weights in a particular generative model. The generative story of this model is simply as follows:

$$\begin{aligned} z &\sim \mathcal{N}(0, I_k), \min(p, q) \leq k \leq 1 \\ x|z &\sim \mathcal{N}(W_X z + \mu_X, \psi_X), W_X \in \mathbb{R}^{p \times k} \\ y|z &\sim \mathcal{N}(W_Y z + \mu_Y, \psi_Y), W_Y \in \mathbb{R}^{q \times k} \end{aligned} \tag{2.13}$$

where z is the latent vector representation for an example, W_X and W_Y are weight matrices mapping this embedding to observed views, and x and y are the observed views of this example. ψ_X, ψ_Y are positive definite covariance matrices and μ_X, μ_Y are arbitrary means that parameterize independent noise in each view. Bach and Jordan (2005) show that the maximum likelihood estimates of W_X and W_Y are:

$$\begin{aligned} \hat{W}_X &= C_{XX} U M \\ \hat{W}_Y &= C_{YY} V M \end{aligned} \tag{2.14}$$

Here, the C_{XX} and C_{YY} are the sample auto-covariance matrices, U and V are the left singular vectors of C_{XX} and C_{YY} respectively, and M is the square root of the diagonal matrix of singular values $C_{XX}^{-\frac{1}{2}} C_{XY} C_{YY}^{-\frac{1}{2}}$.

This probabilistic model demonstrates when CCA is appropriate for learning a representation: *when variation in observed views are independent conditional on their latent representation with independent Gaussian noise applied to each view*. The model also suggests that the CCA problem can be approximately solved by iterative algorithms for estimating latent variables in probabilistic models such as Expectation Maximization.

2.2.3 Nonlinear Variants

One drawback of CCA is it can only uncover linear relationships between views. Although less work has been devoted to maximizing correlation between views after nonlinear transformation, there are two prominent methods for doing so: *Kernel CCA* and *Deep CCA*.

2.2.3.1 Kernel CCA

One method to uncover a nonlinear relationship between two views is to consider kernel CCA (Lai and Fyfe, 2000) (KCCA). Similar to kernel PCA, the practitioner defines kernel functions that independently define the similarity between points in view 1 and view 2, and these kernel functions are used in lieu of inner product when computing correlation.

Problem Let $K_X \in \mathbb{R}^{n \times n}$ and $K_Y \in \mathbb{R}^{n \times n}$ be symmetric positive semi-definite *Gram* matrices expressing the similarity between examples according to features from views 1 and 2. The kernel CCA problem is defined as:

$$\begin{aligned} \max_{z_X \in \mathbb{R}^n, z_Y \in \mathbb{R}^n} corr(z_X, z_Y) = \langle z_X, z_Y \rangle = \alpha^T K_X^T K_Y \beta \\ \text{where } z_X = K_X \alpha, z_Y = K_Y \beta \\ \text{subject to } \|z_X\|_2 = \sqrt{\alpha K_X^2 \alpha} = 1 \\ \|z_Y\|_2 = \sqrt{\beta K_Y^2 \beta} = 1 \end{aligned} \tag{2.15}$$

where $\alpha, \beta \in \mathbb{R}^n$ take the place of the canonical weights u, v in vanilla CCA. Note also that K_X and K_Y replace X and Y. The similarity between this problem formulation and linear CCA comes from the fact that the canonical variates, z_X and z_y, lie in the span of the data in both problems.

Derivation Similar to the Hotelling solution, we can form the augmented Lagrangian, take the derivative with respect to the weights α and β, and set them equal to zero:

$$\begin{aligned} \mathcal{L}(\alpha, \beta, \lambda_1, \lambda_2) &= \alpha^T K_X^T K_Y^T - \lambda_X(\alpha^T K_X^2 \alpha - 1) - \lambda_Y(\beta^T K_Y^2 \beta - 1) \\ \frac{\delta \mathcal{L}}{\delta \alpha} &= K_X^T K_Y \beta - 2\lambda_X K_X^2 \alpha = 0 \\ \frac{\delta \mathcal{L}}{\delta \beta} &= K_Y^T K_X \alpha - 2\lambda_Y K_Y^2 \beta = 0 \end{aligned}$$

We then right-multiply each derivative by α^T and β^T respectively, substitute in the unit-norm constraints, and find that at the solution $\lambda_X = \lambda_y$:

$$\begin{aligned}
\alpha^T K_X^T K_Y \beta - 2\lambda_X \alpha^T K_X^2 \alpha &= 0 \\
\beta^T K_Y^T K_X \alpha - 2\beta^T \lambda_Y K_Y^2 \beta &= 0 \\
2\lambda_X \alpha^T K_X^2 \alpha &= 2\lambda_Y \beta^T K_Y^2 \beta \\
2\lambda_X &= 2\lambda_Y = 2\lambda;
\end{aligned}$$

If we substitute λ back in and solve for α in $\frac{\delta \mathcal{L}}{\delta \alpha} = 0$, we find that:

$$\begin{aligned}
K_X^T K_Y \beta - 2\lambda K_X^2 \alpha &= 0 \\
\frac{K_X^T K_Y \beta}{2\lambda} &= K_X^2 \alpha \\
\frac{K_X^{-1} K_Y \beta}{2\lambda} &= \alpha
\end{aligned}$$

Substituting for α in $\frac{\delta \mathcal{L}}{\delta \beta} = 0$ yields:

$$\begin{aligned}
K_Y^T K_x (\frac{K_X^{-1} K_Y \beta}{2\lambda}) &= 2\lambda K_Y^T K_Y \beta \\
K_Y^2 \beta &= 4\lambda^2 K_Y^2 \beta
\end{aligned}$$

This suggests that if K_Y is invertible, then β is completely unconstrained, and the canonical correlation is 1. A standard solution is to change the unit-norm constraints to be $\alpha^T (K_X + \epsilon_X I)\alpha = 1$ and $\beta^T (K_X + \epsilon_Y I)\beta = 1$. The choice of regularization strength can be selected by heldout correlation captured. This is

Details KCCA can be applied to test data by computing the similarity between each test example to each training example, then mapping this kernel matrix with α or β depending on the view[3]. For example, say we compute $K_X^{test} \in \mathbb{R}^{n \times m}$ for m test examples:

[3]This is analogous to the way one applies kernel PCA to test data.

$$z_X^{test} = \alpha K_X^{test}$$

The derivation is instructive since we see how closely the kernel CCA derivation follows the original Hotelling solution, and it is important since it underscores the necessity of regularizing the Gram matrix regularization (otherwise the problem is not well-defined). This idea of regularization is also applicable to the original CCA objective, where a small amount of diagonal weight can be added to the sample auto-covariance matrices, mostly to ensure invertibility.

2.2.3.2 Deep CCA

Although KCCA maximizes correlation between views subject to an implicit nonlinear mapping, the nonlinear mapping solely depends on the choice of kernel and the training set. In addition, computing and inverting the Gram matrices are very expensive operations in space and computation time. One model that solves these problems is Deep CCA (DCCA), a CCA variant that alternates between maximizing correlation between views and updating a nonlinear mapping from observed views to shared space (Andrew et al., 2013). The nonlinear mappings for each view are parameterized by two neural networks. In addition to *learning* the nonlinear mappings to shared space, DCCA also avoids computing and inverting large $n \times n$ Gram matrices. The crux of fitting DCCA to a dataset lies in the gradient update to update the per-view neural networks.

Problem The DCCA problem for the first canonical component is defined as follows:

$$\begin{aligned} &\max_{\theta_X, \theta_Y, u, v} z_X^T z_Y \\ \text{where } &z_X = f_X(X; \theta_X)u \\ &z_Y = f_Y(X; \theta_Y)v \\ \text{subject to } &\|z_X\|_2 = 1 \\ &\|z_Y\|_2 = 1 \end{aligned} \tag{2.16}$$

Here θ_X and θ_Y are the set of weights parameterizing fixed neural network architectures f_X and f_Y respectively. These are functions that map each example view to a fixed vector of the dimensionality of the network output layer p_f and q_f.

Note that if the network weights are fixed, the solution for canonical weights u and v is just that given by linear CCA with respect to the output layer activations on the training set. In addition, if we fix the canonical weights then we can update network weights θ_X and θ_Y by backpropagation (assuming we can differentiate the objective with respect output layer activations $f_X(X; \theta_X)$ and $f_Y(Y; \theta_Y)$).

This suggests an optimization scheme where at each iteration we alternate between updating network weights by backpropagation and then solving for canonical weights. This way the orthonormality constraint on canonical components is maintained after each iteration.

DCCA Gradient Let the output layer activations of two views passed through their associated networks be X_f and Y_f, and assume that each has zero mean. The gradient of the correlation objective with respect to X_f is given as:

$$\begin{aligned} \frac{\delta corr(X_f, Y_f)}{\delta X_f} \propto & \nabla_{XY} Y_f - \nabla_{XX} X_f \\ \text{where } \nabla_{11} = & C_{XX}^{-\frac{1}{2}} U D U^T C_{XX}^{-\frac{1}{2}} \\ \nabla_{12} = & C_{XX}^{-\frac{1}{2}} U V^T C_{YY}^{-\frac{1}{2}} \end{aligned} \tag{2.17}$$

The partial derivative with respect to Y_f is similar. Note that here we overload the notation for C_{XX} and C_{YY} to be the sample auto-covariance matrices of the *output layer activations*, X_f and Y_f. Similarly C_{XY} is the cross-covariance matrix of X_f and Y_f. U, Λ, and V are solved by a singular value decomposition in the Ewerbring (1990) CCA solution: $C_{XX}^{-\frac{1}{2}} C_{XY} C_{YY}^{-\frac{1}{2}} = \widetilde{U} \Lambda \widetilde{V}^T$ (equation 2.11).

2.2.4 (Many-View) Generalized Canonical Correlation Analysis

Can we apply CCA to maximize correlation between more than just two views? Unfortunately there is no single generalization of correlation to more than a pair of random variables. The extensions to more than two views, generalized CCA (GCCA),

frame the multiview correlation analysis problem as one of optimizing some function of the correlation matrix between all pairs of views, ϕ:

$$\phi = \begin{bmatrix} z_1^T z_1 & z_1^T z_2 & \cdots & z_1^T z_V \\ z_2^T z_1 & z_2^T z_2 & \cdots & z_2^T z_V \\ \vdots & \vdots & \ddots & \vdots \\ z_v^T z_1 & z_V^T z_2 & \cdots & z_V^T z_V \end{bmatrix} \tag{2.18}$$

$$\text{where } \forall i \in 1 \ldots z_i = X_i u_i$$

$$\text{subject to } \|z_i\| = 1$$

where V is the number of views in our data, $X_i \in \mathbb{R}^{n \times p_i}$ is the data matrix, $u_i \in \mathbb{R}^{p_i}$ are the canonical weights, and $z_i \in \mathbb{R}^n$ are the canonical variates for view i.

2.2.4.1 Problem Formulations

Kettenring (1971) gives five different formulations of linear many-view CCA objective. In these formulations, the canonical variates are learned, one-at-a-time, with the constraint that the variates are orthogonal to each other. For each canonical variate, we want to find the canonical weights u_i that satisfy one of the following optimization problems:

- SUMCOR: maximize the sum of correlations: $\max \sum_{i,j \in 1 \ldots V} \phi_{i,j}$
- MAXVAR: maximize the largest eigenvalue of ϕ: $\max \lambda_1$
- SSQCOR: maximize sum of squared correlations: $\max \sum_{i,j \in 1 \ldots V} \phi_{i,j}^2$
- MINVAR: minimize the smallest eigenvalue of ϕ: $\min \lambda_V$
- GENVAR: minimize determinant of ϕ: $det(\phi) = \prod_{i \in \ldots V} \lambda_i$

In this thesis we focus on the MAXVAR objective to learn user embeddings[4]. The MAXVAR formulation is attractive since the optimal canonical weights U_i can be found by standard linear algebra operations and singular value decompositions, much like the two-view CCA objective.

2.2.4.2 MAXVAR GCCA Problem

Kettenring (1971) shows that the MAXVAR GCCA formulation is equivalent to a problem presented earlier by Carroll (1968). This formulation introduces an auxiliary variable to the problem, a matrix $G \in \mathbb{R}^{n \times k}$ that acts as a low-dimensional shared representation across views. The MAXVAR objective with auxiliary variable formulation is as follows:

$$\begin{gathered} \arg\min_{G,U_i} \sum_i \|G - X_i U_i\|_F^2 \\ \text{such that } G^T G = I^k \end{gathered} \tag{2.19}$$

Assuming columns of each X_i are centered, the optimal solution for shared representation G and canonical weights U_i is given as:

$$\begin{gathered} G\Lambda G = \sum_{i=1}^{V} X_i (X_i^T X_i)^{-1} X_i^T \text{eigendecomposition of rhs} \\ \forall i \in 1 \dots V, U_i = (X_i^T X_i)^{-1} X_i^T G \end{gathered} \tag{2.20}$$

Multiview LSA Unfortunately, the MAXVAR GCCA solution given above does not scale well as the number of examples in one's dataset nor the dimensionality of each view increases. Note that the matrix whose eigenvectors are G has n rows and

[4]In Chapter 3, subsection 3.3.2 we also consider an iterative algorithm to approximately solve for weights satisfying the SUMCOR objective .

columns. As the number of examples increases, this matrix will quickly become impossible to store in RAM on most computers. Similarly, inverting the $p_i \times p_i$ auto-covariance matrix, $(X_i^T X_i)$, will quickly become intractable as the dimensionality of view i increases.

Rastogi, Van Durme, and Arora (2015) offers several important tweaks to make this solution tractable: the Multiview LSA algorithm. The key contribution of Multiview LSA is that they consider a truncated SVD decomposition of each view's data matrix: $X_i = A_i S_i B_i^T$. They use these low-rank decompositions to avoid forming the full $n \times n$ matrix in 2.20. They show that G is approximately the left singular vectors of the following matrix:

$$\begin{pmatrix} A_1 & \cdots & A_V \end{pmatrix}$$

In practice, they also regularize the auto-covariance matrices, which leads to a slight scaling of the columns of each A_i.

Details One mild assumption for these methods is that the covariance matrices be invertible, that they have full rank. This assumption can be fulfilled by adding a small value to the diagonal of each of the covariance matrices. For example, $\widetilde{C}_{XX} = C_{XX} + \epsilon I^{p \times p}$. This tweak is known as Regularized CCA or canonical ridge when applied to the two-view CCA problem and is a critical component in Multiview LSA.

One potential difficulty with this solution are the orthonormality constraints on the columns of G. *This constraint applies to all examples across the entire dataset*. Because of this, it is not clear how one would design a stochastic or minibatch

algorithm to solve the MAXVAR GCCA problem that only considers a subset of examples at a time.

It is also important to consider the presence of missing data within views when applying GCCA to real data. Van De Velden and Bijmolt (2006) introduces masking matrices $\forall i \in 1 \dots V, K_i$ into the MAXVAR objective to address this problem:

$$\underset{G,U_i}{\arg\min} \sum_i \|K_i(G - X_i U_i)\|_F^2 \tag{2.21}$$

Each mask, $K_i \in \mathbb{R}^{n \times n}$, is a diagonal matrix, where diagonal elements are either 0 or 1. Examples with data missing in a view are encoded by a zero whereas views with data present are encoded by a one. Although this is cosmetic, it is important to include these masking term, otherwise canonical weights will be artificially forced to map views towards zero (assuming views with missing data are represented as zero vectors).

2.2.4.3 Neural Alternatives to GCCA

Neural architectures that maximize a correlation objective are popular alternatives and scale better to large numbers of examples than classic solutions to GCCA problems. Kumar, Rai, and Daume (2011) elegantly outlines two main approaches these methods take to learn a joint representation from many views: either by (1) explicitly maximizing pairwise similarity/correlation between views or by (2) alternately optimizing a shared, "consensus" representation and view-specific transformations to maximize similarity.

Models such as the Siamese network proposed by Masci et al. (2014), fall in the former camp, minimizing the squared error between embeddings learned from each view, leading to a quadratic increase in the terms of the loss function size as

the number of views increase. Rajendran et al. (2015) extends Correlational Neural Networks (Chandar et al., 2015) to many views and avoid this quadratic explosion in the loss function by only computing correlation between each view embedding and the embedding of a pivot view. Although this model may be appropriate for tasks such as multilingual image captioning, there are many datasets where there is no clear method of choosing a pivot view. The MAXVAR-GCCA objective does not suffer from a quadratic increase in computational complexity with respect to the number of views, nor does it require a privileged pivot view, since the shared representation is learned from the per-view representations.

2.2.4.4 Nonlinear (Deep) GCCA

In spite of encouraging theoretical guarantees, multiview learning techniques cannot freely model nonlinear relationships between arbitrarily many views. Either they are able to model variation across many views, but can only learn linear mappings to the shared space (Horst, 1961), or they simply cannot be applied to data with more than two views using existing techniques based on kernel CCA (Hardoon, Szedmak, and Shawe-Taylor, 2004) and deep CCA (Andrew et al., 2013). Deep Generalized Canonical Correlation Analysis (*dGCCA*) is one recently-introduced model that fills this gap. Here we briefly describe the *dGCCA* model – see Benton et al. (2017) for further details.

Model *dGCCA* is a model that can benefit from the expressive power of deep neural networks and can also leverage statistical strength from more than two views in data, unlike Deep CCA which is limited to only two views.

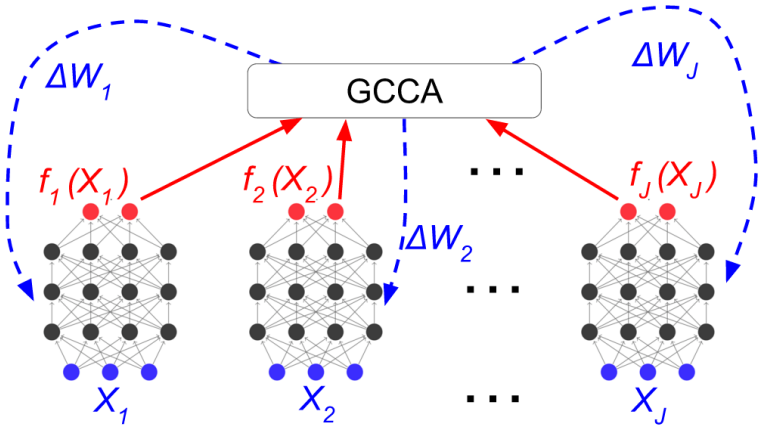


Figure 2.1: A schematic of DGCCA with deep networks for J views.

dGCCA learns a nonlinear map for each view in order to maximize the correlation between the learnt representations across views. In training, *dGCCA* passes the input vectors in each view through multiple layers of nonlinear transformations and backpropagates the gradient of the GCCA objective with respect to network parameters to tune each view's network, as illustrated in Figure 2.1. The objective is to train networks that reduce the GCCA reconstruction error among their outputs. At test time, new data can be projected by feeding them through the learned network for each view.

In the *dGCCA* problem, we consider J views in our data and let $X_j \in \mathbb{R}^{d_j \times N}$ denote the j^{th} input matrix.[5] The network for the j^{th} view consists of K_j layers. Assume, for simplicity, that each layer in the j^{th} view network has c_j units with a final (output) layer of size o_j.

The output of the k^{th} layer for the j^{th} view is $h_k^j = s(W_k^j h_{k-1}^j)$, where $s : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function and $W_k^j \in \mathbb{R}^{c_k \times c_{k-1}}$ is the weight matrix for the k^{th} layer of the j^{th} view network. We denote the output of the final layer as $f_j(X_j)$.

[5]Our notation for this section closely follows that of Andrew et al. (2013)

dGCCA can be expressed as the following optimization problem: find weight matrices $W^j = \{W_1^j, \ldots, W_{K_j}^j\}$ defining the functions f_j, and linear transformations U_j (of the output of the j^{th} network), for $j = 1, \ldots, J$, such that

$$\begin{aligned} &\underset{U_j \in \mathbb{R}^{o_j \times r}, G \in \mathbb{R}^{r \times N}}{\arg\min} \sum_{j=1}^{J} \|G - U_j^\top f_j(X_j)\|_F^2 \\ &\text{subject to} \qquad GG^\top = I_r \end{aligned}, \tag{2.22}$$

where $G \in \mathbb{R}^{r \times N}$ is the shared representation we are interested in learning.

Gradient Derivation Sketch Next, we show a sketch of the gradient derivation. See Benton et al. (2017) for the full gradient derivation with respect to network output layer. It is straightforward to show that the solution to the GCCA problem is given by solving an eigenvalue problem. In particular, define $C_{jj} = f(X_j)f(X_j)^\top \in \mathbb{R}^{o_j \times o_j}$ to be the scaled empirical covariance matrix of the j^{th} network output, and let $P_j = f(X_j)^\top C_{jj}^{-1} f(X_j) \in \mathbb{R}^{N \times N}$ be the corresponding projection matrix that whitens the data; note that P_j is symmetric and idempotent. We define $M = \sum_{j=1}^{J} P_j$. Since each P_j is positive semi-definite, so is M. Then, it is easy to check that the rows of G are the top r (orthonormal) eigenvectors of M, and $U_j = C_{jj}^{-1} f(X_j) G^\top$. Thus, at the minimum of the objective, we can rewrite the reconstruction error as follows:

$$\sum_{j=1}^{J} \|G - U_j^\top f_j(X_j)\|_F^2 = \sum_{j=1}^{J} \|G - G f_j(X_j)^\top C_{jj}^{-1} f_j(X_j)\|_F^2 = rJ - \mathrm{Tr}(GMG^\top)$$

Minimizing the GCCA objective (w.r.t. the weights of the neural networks) means maximizing $\mathrm{Tr}(GMG^\top)$, which is the sum of eigenvalues $L = \sum_{i=1}^{r} \lambda_i(M)$. Taking

the derivative of L with respect to each output layer $f_j(X_j)$ we have:

$$\frac{\partial L}{\partial f_j(X_j)} = 2U_j G - 2U_j U_j^\top f_j(X_j)$$

Thus, the gradient is the difference between the r-dimensional auxiliary representation G embedded into the subspace spanned by the columns of U_j (the first term) and the projection of the actual data in $f_j(X_j)$ onto the said subspace (the second term). Intuitively, if the auxiliary representation G is far away from the view-specific representation $U_j^\top f_j(X_j)$, then the network weights should receive a large update. Computing the gradient descent update has time complexity $O(JNrd)$, where $d = max(d_1, d_2, \ldots, d_J)$ is the largest dimensionality of the input views.

Relationship to Semi-supervised Models over Text and Network Rahimi, Cohn, and Baldwin (2018) describe applying a graph convolutional network (GCN) to predict Twitter user location. Although this model merges text and network information about users it is not directly related to the multiview methods we analyze here. The GCN is fundamentally a semi-supervised model that differentially weights features of neighboring users within the network graph to better infer node labels. On the other hand, multiview methods learn example representations by finding transformations of each example view to maximize the between-view correlation rather than predicting supervision.

However, there is nothing preventing one from using a GCN as a view transformation layer, which can subsequently be tuned according to a *dGCCA* objective. The *dGCCA* learning algorithm is agnostic to the architecture of the transformation network as well as the form of input view representation, so long as the objective is differentiable with respect to the neural network weights.

2.3 Multitask Learning and Neural Models

In chapters 5 and 6, we use a machine learning framework called *multitask learning* (MTL) to inject user information into classification models. In this section we present the MTL setting at a high level and discuss why neural networks are particularly convenient models to train in this framework.

2.3.1 Motivation

MTL was first presented in Caruana (1993) and is discussed in detail in Caruana's dissertation (Caruana, 1997). MTL is a machine learning framework for exploiting *related* auxiliary tasks to improve a classifier's generalization performance at some main task that the practitioner cares about. The classifier is trained to perform well according to these auxiliary tasks along with the main task, updating weights or a representation common across the tasks. Caruana describes MTL as introducing a human "inductive bias" to the main task model. This inductive bias is encoded by which additional tasks the practitioner believes will serve as useful guides to a model that must perform well at the single main task.

Consider an example from Collobert et al. (2011). In this paper, the authors want to improve a semantic role labeling system. This system takes a sequence of tokens as input and generates a sequence of labels encoding semantic roles, one for each token, as output – this is their main task. They then consider a related task of language modeling – the auxiliary task. The auxiliary language modeling task is formulated as maximizing the score that the model assigns to real English sentences, while minimizing the score assigned to fake, generated English sentences. They hypothesize that a model that can successfully discriminate between real and fake English text

will be better at assigning semantic role labels than models that have a poor sense of what constitutes well-formed English. They report reducing semantic role labeling word error rate from 16.5 to 14.4 after joint MTL training with the language modeling auxiliary task – over a 10% reduction in word error rate.

2.3.1.1 Benefits

There are several benefits to the MTL framework aside from improving classifier generalization over traditional single-task learning.

The first is that auxiliary tasks are only necessary during train, not at test time. The auxiliary tasks serve as beneficial regularizers for the classifier being learned, and can be discarded. This is analogous to the way that neural network weights may be trained with dropout, weight decay, or other regularization techniques, but those terms only influence how weights are updated during training, not how predictions are ultimately made. This was the major motivation behind using multitask learning to improve pneumonia risk prediction given medical history in Caruana, Baluja, and Mitchell (1996). In this work, they use the results from lab tests as auxiliary tasks. These lab tests are time-consuming, expensive, and are only available *after a patient has been hospitalized*. However, they are predictive of pneumonia risk, so a classifier that can predict these results will also better predict pneumonia risk. These lab results are available at train time, but are clearly unavailable during test.

MTL is especially effective when there are few labeled training examples for the main task, but many labeled examples when considering the entire set of auxiliary related tasks. This effectively expands the training size of the classifier's training set, reducing the error incurred by sampling variance. Secondly, MTL allows for datasets where different subsets of examples are annotated for different tasks. This

is especially important since it allows the practitioner to combine disparate datasets togther even though disjoint example sets are annotated in each case.

2.3.2 Learning Setting

In supervised MTL, we want to train a classifier such that it achieves low expected loss across multiple tasks. We are given a total of T tasks and one classifier for each task, $\{f_1, f_2, \ldots f_T\}$. The classifier for task t maps examples from domain $\mathcal{X}$ to predictions in domain $\mathcal{Y}_t$[6]. Each classifier f_t is determined by two sets of parameters:

- Θ: parameters shared by all classifiers
- θ_t: the task-specific parameters for task t

One can also consider all the f_t as a single model that generates vector-valued predictions : $\langle y_1, y_2, \ldots, y_T \rangle$.

If $\mathcal{L}_t : \mathbb{R}^{\mathcal{Y}_t, \mathcal{Y}_t} \rightarrow \mathbb{R}^+$ is the loss function for task t and examples are drawn from the joint probability distribution P_t, then the MTL objective is as follows:

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{x,y \sim P_t} \mathcal{L}_t(y, f_t(x; \Theta, \theta_t)) \tag{2.23}$$

In other words, we want to learn task-shared parameters, Θ, and task-specific parameters, $\{\theta_1, \ldots, \theta_T\}$, that minimize the average expected loss across all tasks[7].

[6]The assumption that the domain of each classifier is the same is actually a simplification. In general, the domain of each classifier may be different (either subsets of a shared feature set or completely different feature sets), so long as there exist parameters shared across classifiers (Zhang and Yeung, 2011).

[7]In practice the objective will also include a regularization term to penalize large parameter weights. This term is orthogonal to the multitask objective, though.

2.3.2.1 Neural Models

Caruana (1993) first introduces MTL in the context of neural models, and for good reason: MTL is trivial to implement in a neural model. This is because neural models are simple to optimize with respect to multiple loss functions, so long as each loss function is differentiable with respect to the model parameters.

Take the model presented in Li, Ritter, and Jurafsky (2015) for learning Twitter user representations in an MTL framework that capture both similarity in posted text and closeness in the social network [8].

Example: Multitask Learning of User Representations A simplified version of their model is diagrammed in Figure 2.2. Consider two tasks to aid learning vector representations for Twitter users: a user-conditioned language modeling task and a friend prediction task. Their language modeling task is described by the following objective:

$$\begin{aligned} C(w_i, u) &= \sum_{j=i-k; j\neq i}^{i+k} L(w_j) \\ s(w_i \| C(w_i), u) &= \frac{C(w_i) + e_u}{2k+1} W_{\text{text}} \\ p(w_i \| C(w_i), u) &= \frac{s(w_i, u)_{w_i}}{\sum s(w_i, u)} \end{aligned} \tag{2.24}$$

where L is a word embedding lookup table mapping word indices to vectors, $C(w_i)$ is the k-window word embedding context around word w_i, and e_u is the vector representation for user u. e_u as well as W_{text} and the word embeddings in L are all

[8] In their full model, Li, Ritter, and Jurafsky (2015) also learn representations that are predictive of other attributes such as occupation, location, and gender as well. We omit these additional tasks as we mean to use this as a simple model of how neural network models and training is especially conducive to MTL. Inferring user representations is also a key problem in this thesis.

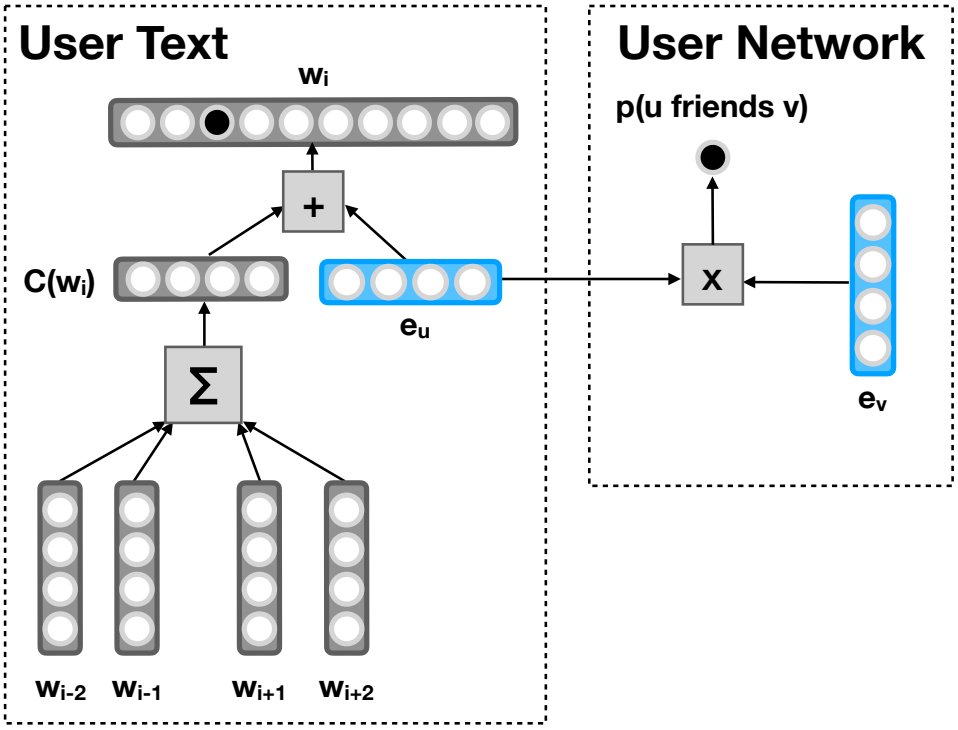


Figure 2.2: Diagram of two tasks presented in Li, Ritter, and Jurafsky (2015) to learn user representations in an MTL setting. The user representations that are learned are depicted by the blue vectors e_u and e_v, and components for the text modeling and friendship prediction tasks are separated by dotted lines. The text prediction task is addressed by a multinomial logistic regression model – the feature set is the mean of average context word embeddings and a user's vector representation. The friendship prediction task is defined as a parameterless logistic regression model determined solely by the dot product of the two user representations.

parameters learned during model training to maximize the probability, $p(w_i \| C(w_i), u)$. This is equivalent to the Paragraph Vector embedding technique where instead of learning a vector for each paragraph, we learn a representation for each user (Le and Mikolov, 2014). The model is trained similarly to paragraph vectors, where positive examples is the instantiated word in its context and negative examples are generated by sampling uniformly from the remainder of the vocabulary.

The friend prediction task is modeled more simply:

$$p(u \text{ friends } v) = \frac{1}{1 + exp\{-e_u^T e_v\}} \tag{2.25}$$

where e_u and e_v are embeddings for two distinct users u and v. The probability that user u is friends with v is determined by passing their dot product through a sigmoid function.

The empirical log-likelihood for both of these tasks for a set of users U, the sequence of words each user posts w^u, and pairs of friends F is then:

$$\begin{aligned} \log \mathcal{L}_{\text{text}} &= \sum_{u \in U} \sum_{w_i \in w^u} p(w_i \| C(w_i), u) \\ \log \mathcal{L}_{\text{friend}} &= \sum_{(u,v) \in F} p(u \text{ friends } v) \\ \log \mathcal{L}_{\text{joint}} &= \mathcal{L}_{\text{text}} + \mathcal{L}_{\text{friend}} \end{aligned} \tag{2.26}$$

Model parameters can be learned by alternately sampling pairs of users for the friend prediction task, and sampling words in context for the language modeling task to update user representations. For each task, user representations are updated by stochastic gradient descent. This is a consequence of the fact that the joint loss for

both of these tasks is a linear combination of the per-task losses, and so the joint loss remains differentiable with respect to the user representations.

Other Multitask Models Although this thesis only considers neural networks trained in multitask fashion, a wide variety of machine learning models have been extended to the MTL setting.

Along with neural networks, Caruana (1997) presents MTL extensions for feature-weighted k-nearest neighbor regression and decision trees. In this model, the per-feature weights for computing ℓ_2 distance are selected to minimize average mean squared error across tasks rather than for a single task. Multitask decision trees are learned by not just maximizing information gain of a single task, but a weighted average of information gain across all tasks.

Evgeniou and Pontil (2004) present a generic regularization-based framework inspired by fitting support vector machines. In this framework, each task's model is assumed to have parameters which are close to each other. This "closeness" is enforced by penalizing large deviations from a set of shared parameters while maximizing the margin from decision boundary (in the case of max-margin classification). MTL has also influenced learning of unsupervised tasks such as clustering (Gu and Zhou, 2009) and nonlinear regression models such as Gaussian processes (Yu, Tresp, and Schwaighofer, 2005).

2.3.2.2 Options for MTL

The MTL framework is extremely flexible in how models can be trained. This is both a blessing and a curse: flexibility means that MTL is applicable to improving just

about any predictive model, but means that there is potentially more space to explore to find the most effective way to deploy MTL.

Task Selection and Weighting How does one define *related* tasks? This is the fundamental problem in deriving generalization improvements from MTL training and one without a clear answer. Simple measures such as correlation between class labels are not necessary to identify related tasks as (Caruana, 1997). The best definition of task-relatedness offered in Caruana (1997) is not particularly illuminating:

> The most precise definition for relatedness we have been able to devise so far is the following: Tasks A and B are related if there exists an algorithm M such that M learns better when given training data for B as well, and if there is no modification to M that allows it to learn A this well when not given the training data for B. While precise, this definition is not very operational.

This cannot be easily used as a heuristic for task selection since the definition of relatedness amounts to going ahead and training a model on both tasks (showing tasks are unrelated is even more difficult, requiring a sweep over all learning algorithms/-models). Similar to selecting tasks, it is typical to weight auxiliary tasks differently within the loss function based on which are believed to be the most beneficial. This is akin to a soft selection of auxiliary tasks.

Although there are methods to jointly infer how "related" tasks are as well as learn weights for each task (Bakker and Heskes, 2003; Kang, Grauman, and Sha, 2011), there is no panacea, since they typically make strong assumptions on how data were generated. Ultimately, which auxilliary tasks that will lead to best improve

generalization performance must be selected by the practitioner based on their domain knowledge, the data and tasks available, and empirical evaluation.

Training Regimen Practically, there are a few additional decisions that must be made when training neural models. When updating parameters stochastic gradient descent, how should we sample examples? Sampling uniformly at random is not ideal when the number of examples is unbalanced across tasks. It is typical to experiment with how auxiliary tasks should be sampled to avoid entraining weights too strongly towards one or another. In this sense, deciding how to sample examples is similar to deciding how to weight the loss – oversampling examples from one task will lead to parameter updates that improve that task more. A final pass of fine-tuning toward the main task is often helpful. This, however, may be liable to discarding the benefits of MTL if one does not freeze shared parameters or limit the number and size of single-task updates.

2.3.3 Discussion: Relationship to Multiview Methods

The models we consider in this thesis integrate auxiliary information into embeddings and models to improve generalization performance at some downstream task. The integration of auxiliary information can come in the form of a multiview method, learning embeddings that capture correlation between views, supervision used to condition distribution over topics in a topic model, or as an auxiliary task for a supervised classifier.

Since many of the approaches we present here are naturally extended to semi-supervised learning settings (multitask learning in particular), one might be tempted to think of these methods as *transductive*. However, unlike transductive algorithms

which infer labels for a batch of unlabeled examples at training time (Gammerman, Vovk, and Vapnik, 1998), the multiview and multitask algorithms we present here do not need to infer labels for unlabeled examples. Multiview algorithms have no notion of a target or label which the embedding is entrained to predict (although they can easily be extended to have one (Wang et al., 2015)). Multitask training (particularly for neural models) can easily be extended to incorporate new examples labeled for any of the main or auxiliary tasks – main task labels can be inferred by the model, but they need not be used in the training process.

It is more useful to view multiview and multitask learning approaches as ways of inserting inductive biases into learned model features. The major difference between these two approaches, multiview and multitask learning, is in the kinds of biases that each can express. Multiview methods suppose that the latent features one wants to learn are best captured by that which is common between a set of "auxiliary" features, or views. Under this interpretation, we presume that observed views are generated conditional on this latent feature (consider the probabilistic interpretation of CCA (Bach and Jordan, 2005)). In neural multitask learning, one supposes that this latent feature vector (e.g. a hidden layer in one's network) is *predictive of* the auxiliary tasks, and is *generated by* a separate set of input features.

To take an example from chapter 3, suppose we have collected the past tweets and list of local network friends for a large set of Twitter users. We can take two separate approaches to model these users. We could apply a multiview representation learning method such as CCA to map the text and network views to a shared space. This approach assumes that the text and network features are generated independently conditioned on some latent feature vector. If we took a supervised multitask learning approach, we would either predict a user's local network from their text, their text from

their local network, or would predict both text and local network from a completely separate set of input features.

One critical difference between multiview and multitask learning is the flexibility afforded by multitask learning setting. The multiview methods we consider in this thesis are constrained to maximizing correlation between views, and often make strong assumptions on the distribution of observations (e.g. observations are Gaussian distribution). On the other hand, multitask learning is closer to a philosophy of model training that happens to be easily translated to neural network training.

Chapter 3

Multiview Embeddings of Twitter Users

In this chapter we present methods to learn *unsupervised* embeddings for a general set of users from different views of their online behavior. We evaluate these embeddings both intrinsically according to how well they capture hashtag usage and friending behavior and extrinsically according to how well they predict demographic features. This chapter was adapted mainly from Benton, Arora, and Dredze (2016), published as a short paper in ACL 2016. The deep GCCA experiments were presented in Benton et al. (2017), an arχiv preprint. The LasCCA algorithm was implemented and adapted to support missing views during a 2017 summer internship at Amazon.

Dense, low-dimensional vector embeddings have a long history in NLP, and recent work on neural models have provided new and popular algorithms for training representations for word types (Mikolov et al., 2013a; Faruqui and Dyer, 2014), sentences (Kiros et al., 2015), and entire documents (Le and Mikolov, 2014). These embeddings exhibit desirable properties, such as capturing some aspects of syntax or semantics and outperforming their sparse counterparts at downstream tasks.

While there are many approaches to generating embeddings of *text*, it is not clear how to learn embeddings for social media *users*. There are several different types of data (views) we can use to build user representations: the text of messages they post, neighbors in their local network, articles they link to, images they upload, etc. Although user embeddings can always be finetuned for a supervised objective, it is unclear which unsupervised views and methods perform best across a variety of tasks.

Multiview embedding methods such as Generalized Canonical Correlation Analysis (*GCCA*) (Carroll, 1968; Van De Velden and Bijmolt, 2006; Arora and Livescu, 2014; Rastogi, Van Durme, and Arora, 2015) are attractive methods for simultaneously capturing information from multiple user views. These methods may be more appropriate for learning user embeddings than concatenating views into a single vector, since views may correspond to different modalities (image vs. text data) or have very different distributional properties. Treating all features as equal in this concatenated vector would not be appropriate.

In this chapter we present an extension of the MAXVAR-GCCA problem that offers increased flexibility in learning user embeddings than standard *GCCA*: weighted *GCCA* (*wGCCA*). *wGCCA* allows the practitioner to discriminatively weight the per-view loss, forcing user embeddings to capture variation in some views more closely than others. View weighting is chosen based on either a prior notion of which views will be the most informative or by tuning to improve some downstream metric – this is up to the embedder's discretion. We also consider an algorithm to approximately solve the (linear) SUMCOR-GCCA problem, large-scale CCA (Fu et al., 2016) (*LasCCA*), as another multiview user embedding method. We adapt the *LasCCA* implementation presented in Fu et al. (2016) to support data with missing views (especially important when considering data compiled from social media).

We evaluate multiview embeddings at how well they capture hashtag usage and friending behavior and how well they predict user demographic features. We compare their performance at these tasks to single-view baselines and show that the location of users in embedding space can capture average peoples' notions of what constitutes a similar group of users. This is analogous to how word embeddings capture semantic and syntactic properties of word types.

In Section 3.1 we first describe the different types of user behavior used to learn embeddings and how this dataset was assembled. Sections 3.2 and 3.3 describe the baseline and multiview methods we use to learn embeddings. Section 3.4 describes how embeddings were evaluated and Section 3.5 finally contains both quantitative and qualitative evaluation of user embeddings.

3.1 User Behavior Data

What is the best type of behavior to learn user embeddings on? Although the answer ultimately depends on how these embeddings will be used, some types of user behavior and embedding methods will be more appropriate for a variety of tasks. To answer this question, we assembled a dataset of general Twitter users, with multiple aspects of user behavior. Knowing how the dataset was assembled is critical to understanding what kind of user behavior is available to each embedding method.

3.1.1 Data Collection

We uniformly sampled 200,000 users from a stream of publicly available tweets from the 1% Twitter stream from April 2015. We removed users with verified accounts, more than 10,000 followers, or non-English profiles to restrict to typical, English

speaking users[1]. For each user we collected their 1,000 most recent tweets, and then filtered out non-English tweets. We removed users without English tweets in January or February 2015, yielding a total of 102,328 users. Although limiting tweets to only these two months restricted the number of tweets we were able to work with, it also ensured that our data are drawn from a narrow time window, controlling for differences in user activity over wide stretches of time. This allows us to learn distinctions between users, and not temporal distinctions of content.

Next, we collect information about the users' friend and mention networks. Specifically, for each user **mentioned** in a tweet by one of the 102,328 users, we collect their 200 most recent English tweets from January and February 2015. Similarly, we collected the 5,000 most recently added friends and followers for each of the 102,328 users. We then sampled 10 friends and 10 followers for each user and collected up to the 200 most recent English tweets for these followers and friends from January and February 2015. Limits on the number of users and tweets per user were imposed so that we could operate within Twitter's API limits[2].

This data supports our evaluation tasks as well as the four sources of behavior/-content for each user: their tweets, tweets of mentioned users, friends, and followers.

3.1.2 User Views

We consider four main views/sources of information about a user. **ego** information as represented by the text in public tweets the user posts, **mention**ed information represented by messages made by people mentioned in a tweet posted by the ego user, **friend** information for those people who the ego user follows, and **follower**

[1]Language identified with the python module `langid` (Lui and Baldwin, 2012).

[2]The Twitter REST API limits on collecting local network information are especially strict. A non-privileged API key can only pull 5,000 friend/follower IDs per minute for a single user.

information for those who follow the ego user. Although there are other views we could have collected (e.g. the user description or image), prior work has shown that these four views are predictive of latent user attributes, and therefore would be useful for learning user embeddings (Volkova, Coppersmith, and Van Durme, 2014b).

Two main representations are considered when constructing views: either text representations or a direct representation of the friend or follower IDs.

3.1.2.1 Text

For each text source we can aggregate the many tweets into a single document, e.g. all tweets written by accounts mentioned by a user. We represent this document as a bag-of-words (*BOW*) in a vector space model with a vocabulary of the 20,000 most frequent word types after stopword removal. We consider TF-IDF weighted *BOW* vectors. This was done for tweets made by the **ego** user, **mentions**, **friends**, and **followers**.

A common problem with these representations is that they suffer from the curse of dimensionality. A natural solution is to apply a dimensionality reduction technique to find a compact representation that captures as much information as possible from the original input. Here, we consider principal components analysis (PCA), a ubiquitous linear dimensionality reduction technique. The text views that are fed into multiview embedding methods are all first reduced by PCA before learning the embedding. We run PCA and extract up to the top 1,000 principal components for each of the above views. This speeds up fitting multiview embedding methods since the feature dimensionality of each view is reduced.

3.1.2.2 Network

An alternative to text based representations is to use the social network of users as a representation. We encode a user's social network as a vector by treating the set of users in the social graph as a vocabulary, where users with similar social networks have similar vector representations (*NetSim*). This is an n-dimensional vector that encodes the user's social network as a bag-of-words over this vocabulary. In other words, a user is represented by a summation of the one-hot encodings of each neighboring friend or follower in their social network. In this representation, the number of friends two users have in common is equal to the dot product between their social network vectors. We define the social network as one's followers or friends. The motivation behind this representation is that users who have similar networks may behave in similar ways. Such network features are commonly used to construct user representations as well as to make user recommendations (Lu, Lam, and Zhang, 2012; Kywe et al., 2012).

The binary representations of local network are reduced to the top 1,000 principal components, as are the text representations.

3.2 Baseline Embedding Methods

Each of these views can be treated as a user embedding in their own right. They can also be combined using different methods to yield aggregate user representations across views. Here we describe baseline user embeddings we evaluate.

3.2.1 PCA

For the following experiments, we consider the PCA representations as a baseline. We consider up to the top 1,000 principal components within each view as the user

embedding. In order to fairly compare multiview embedding methods to methods that do not maximize correlation between views, we also consider a naïve combination of PCA views as an embedding.

We consider *all* possible combinations of views obtained by concatenating original view features, and subsequently reducing the dimensionality by PCA. By considering all possible concatenation of views, we ensure that this method has access to the same information as multiview methods. Both the raw *BOW* and *BOW-PCA* representations have been explored in previous work for demographic prediction (Volkova, Coppersmith, and Van Durme, 2014b; Al Zamal, Liu, and Ruths, 2012) and recommendation systems (Abel et al., 2011; Zangerle, Gassler, and Specht, 2013). Only the best performing view subset evaluated on the development set is reported on test.

3.2.2 Word2Vec

BOW-PCA is limited to linear representations of *BOW* features based on global context. Modern neural network based approaches to learning word embeddings, including word2vec continuous bag of words and skipgram models, can learn representations that capture local context around each word (Mikolov et al., 2013b). We represent each view as the simple average of the word embeddings for all tokens within that view (e.g., all words written by the ego user). Word embeddings are learned on a sample of 87,755,398 tweets and profiles uniformly sampled from the 1% Twitter stream in April 2015 along with all tweets and profiles collected for our set of users – a total of over a billion tokens. We use the word2vec tool, select either skipgram or continuous bag-of-words embeddings on dev data for each prediction task, and train for 50 epochs. We use the default settings for all other parameters.

3.3 Multiview Embedding Methods

Here we describe three different methods for learning multiview user embeddings. Each of these multiview embedding methods are evaluated against each other at the tasks described in section 3.4.

3.3.1 MAXVAR-GCCA

We use Generalized Canonical Correlation Analysis (*GCCA*) (Carroll, 1968) to learn a single embedding from multiple views. *GCCA* finds G, U_i that minimize:

$$\arg\min_{G,U_i} \sum_i \|G - X_i U_i\|_F^2 \qquad \text{s.t. } G^T G = I \tag{3.1}$$

where $X_i \in \mathbb{R}^{n \times d_i}$ corresponds to the data matrix for the ith view, $U_i \in \mathbb{R}^{d_i \times k}$ maps from the latent space to observed view i, and $G \in \mathbb{R}^{n \times k}$ contains all user representations (Van De Velden and Bijmolt, 2006).

3.3.1.1 Weighted GCCA

Since each view may be more or less helpful for a downstream task, we do not want to treat each view equally in learning a single embedding. Instead, we weigh each view differently in the objective:

$$\arg\min_{G,U_i} \sum_i w_i \|G - X_i U_i\|_F^2 \text{ s.t. } G^T G = I, w_i \geq 0 \tag{3.2}$$

where w_i explicitly expresses the importance of the ith view in determining the joint embedding. The columns of G are the eigenvectors of $\sum_i w_i X_i (X_i^T X_i)^{-1} X_i^T$,

and the solution for $U_i = (X_i^T X_i)^{-1} X_i^T G$. In our experiments, we use the approach of Rastogi, Van Durme, and Arora (2015) to learn G and U_i, since it is more memory-efficient than decomposing the sum of projection matrices.

We also consider a minor modification of *GCCA*, where G is scaled by the square-root of the singular values of $\sum_i w_i X_i X_i^T$ (*GCCA-sv*). This is inspired by previous work showing that scaling each feature of multiview embeddings by the singular values of the data matrix can improve performance at downstream tasks such as image or caption retrieval (Mroueh, Marcheret, and Goel, 2015). Note that if we only consider a single view, X_1, with weight $w_1 = 1$, then the solution to *GCCA-sv* is identical to the PCA solution for data matrix X_1, without mean-centering.

3.3.2 SUMCOR-GCCA

In addition to the MAXVAR-*GCCA* objective, we also consider another generalization of CCA to more than two views: SUMCOR-*GCCA*. The SUMCOR-*GCCA* problem is given in Equation 3.3:

$$\begin{gathered} \underset{\forall i \in [V], U_i \in \mathbb{R}^{m_i \times k}}{\arg\max} \sum_{i=1}^{V} \sum_{j \neq i} Tr[U_i^T X_i^T X_j U_j] \\ \text{subject to } \forall i \in [V], U_i^T X_i^T X_i U_i = I^k \end{gathered}, \tag{3.3}$$

where V is the number of views and U_i are the canonical weights for view i. SUMCOR-*GCCA* seeks to find mappings that maximize the sum of total correlation captured between every pair of views while ensuring that the canonical variates for each view are orthonormal as in CCA. This differs from the MAXVAR-GCCA objective in two ways: (1) SUMCOR-*GCCA* requires no nuisance variable, G, to ensure views are mapped close to each other. The orthonormality of projected views is

ensured by the hard constraints in the objective. (2) The SUMCOR-*GCCA* problem seeks to maximize the sum of correlations between each pair of views. The MAXVAR formulation instead seeks to maximize the maximum eigenvalue of the correlation matrix between all pairs of views (Kettenring, 1971).

Jointly solving for all U_i is difficult, so we run the Large-scale generalized CCA (*LasCCA*) algorithm (Fu et al., 2016) for a fixed number of iterations (100) to solve for the mappings for each view. *LasCCA* proceeds by maximizing the SUMCOR-*GCCA* objective with respect to each U_i round-robin, holding all other view mappings fixed. We consider this multiview objective for three reasons: (1) It allows us to compare if a slightly different multiview objective yields similarly-performing embeddings to those learned to maximize the MAXVAR-GCCA. (2) We can assess how performant the learned embeddings are as a function of *LasCCA* epochs devoted to solving the GCCA problem. (3) Although *LasCCA* does not guarantee an optimal solution, the algorithm is designed to scale well when the input views are very high-dimensional and sparse, avoiding keeping low-rank approximations to the sum of projection matrices as in multiview LSA. This allows *LasCCA* to learn multiview embeddings directly from, for example, a bag-of-words in all of a user's tweets.

3.3.2.1 Robust *LasCCA* Algorithm

The *LasCCA* algorithm is shown in Algorithm 1, and relies on the subroutine `H_Compute` in Algorithm 2. In order to support our Twitter data, we modified the original *LasCCA* algorithm to ignore views with missing data similar to the modification of multiview LSA. The terms that differentiate this algorithm from the *LasCCA* algorithm presented in Fu et al. (2016) are highlighted in red. *LasCCA* is more computationally efficient than standard GCCA algorithms with many, high-dimensional views. For

Algorithm 1 Rank-k robust *LasCCA*

Require: $\{X_i, G_i, K_i\}_{i=1}^V$ ◁ Observations, auxiliary variates, and masks for each view
Require: T ◁ No. of epochs
for $i \leftarrow 1 \dots V$ **do**
  $\mathcal{K}_i = \sum_{j=1, j\neq i}^V K_j$ ◁ No. of non-zero views per example
  $\mathbb{K}_i = \mathbb{1}[\mathcal{K}_i] K_i$ ◁ Indicator of data in view i *and* at least one other view
end for
for $t \leftarrow 1 \dots T$ **do**
  for $i \leftarrow 1 \dots V$ **do**
    $H_i \leftarrow \texttt{H}_{\texttt{compute}}(\{X_i\}_{i=1}^V, \{G_j\}_{j=1, j\neq i}^V, \mathcal{K}_i, \mathbb{K}_i)$ ◁ See Algorithm 2
    $U_i' S_i V_i' \leftarrow H_i$ ◁ Singular value decomposition of H_i
    $G_i \leftarrow U_i' V_i'$
    $\hat{U}_i \leftarrow \arg\min_U \|\mathbb{K}_i (X_i U - G_i)\|_2^F$
  end for
end for

return $(\{G_i\}_{i=1}^V, \{\hat{U}_i\}_{i=1}^V)$

Algorithm 2 $\texttt{H}_{\texttt{compute}}$ subroutine for *LasCCA*

Require: i ◁ View to calculate H for
Require: $\{X_j\}_{j=1}^V, \{G_j\}_{j=1, j\neq i}^V, \mathcal{K}_i, \{\mathbb{K}_j\}_{j=1}^V$ ◁ Observations, auxiliary variates, and masking matrices for each view
for $j \leftarrow 1 \dots V$ **do**
  if $j \neq i$ **then**
    $\hat{R}_j \leftarrow \arg\min_R \|\mathbb{K}_j (X_j R - G_j)\|_2^F$
    $C_j \leftarrow \mathbb{K}_j X_j \hat{R}_j$
  end if
end for
$P_i \leftarrow \frac{V}{\mathcal{K}_i} \sum_{j=1, j\neq i}^V C_j$
$\hat{E}_i \leftarrow \arg\min_E \|\mathbb{K}_i (X_i E - P_i)\|_2^F$
$H_i \leftarrow \mathbb{K}_i X_i \hat{E}_i$
return H_i

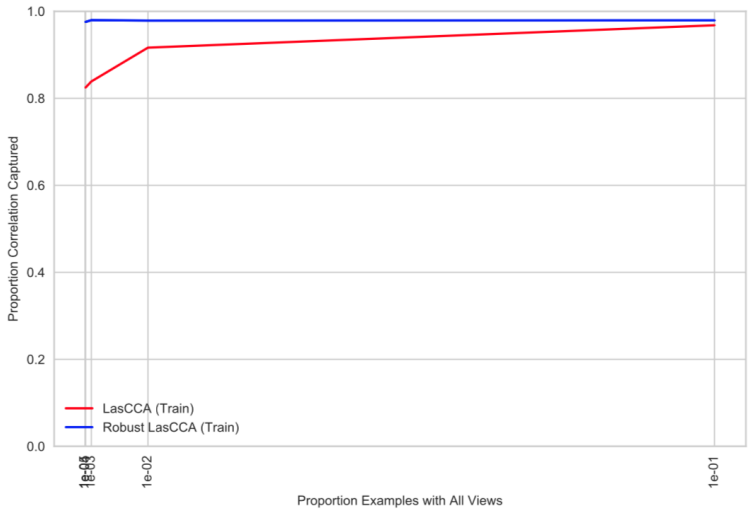


Figure 3.1: Proportion of train correlation captured by vanilla and robust *LasCCA* after 5 epochs of training, learning $k = 10$ canonical variates. Proportion of examples where data from all-views-but-one are missing is listed along the x-axis. The leftmost point corresponds to a dataset where only 10 out of 10^5 examples have active features in all views. Proportion correlation captured of 1.0 is optimal.

example, Rastogi, Van Durme, and Arora (2015) requires (at best) an SVD of a dense $n \times (Vk)$ matrix to solve the MAXVAR *GCCA* objective, where k determines the low-rank approximation of each view's data matrix. *LasCCA* only involves solving a series of (possibly sparse) linear least squares problems, and computing SVDs of dense $n \times k$ data matrices which can be performed in $O(nk^2)$ time.

Robust LasCCA ignores views where no features are active, avoiding unnecessarily entraining projected views toward zero. $\mathbb{K}_i$ is an $n \times n$ diagonal matrix that masks examples where either (1) view i has no active features or (2) *all views but* i have no active features. $\mathcal{K}_i$ encodes the number of active views that are not view i, for each example. This is used to rescale the rows of P_i, to account for examples that are missing data from views.

We verified the correctness of robust *LasCCA* in the face of missing views by recovering directions of maximal correlation from a synthetic dataset. Data was generated by first creating $N = 10^5$ example latent feature vectors of dimension $F = 100$, with a randomly selected five features active in each example, values sampled from a unit Gaussian. $F \times F$ sparse maps from latent to observed views (three views total) were generated with 10% non-zero values, values also drawn from unit Gaussians. A missingness parameter $\rho \in [0, 1]$ was varied, such that ρ proportion of examples contained active features in one and only one view, and $(1 - \rho)$ proportion of examples contained active features in all views. Figure 3.1 shows that this robust extension of *LasCCA* achieves near optimal proportion correlation captured on train, regardless of how many examples are missing data. Vanilla *LasCCA* is sensitive to these missing view examples, artifically forcing the view 1 projection toward zero when these examples should be ignored.

3.4 Experiment Description

We selected three prediction tasks to evaluate the effectiveness of the multi-view user embeddings: user engagement prediction, friend recommendation and demographic characteristics inference. Our focus is to show the performance of multiview embeddings compared to other representations, not on building the best system for a given task.

3.4.1 Learning Embedding Details

GCCA embeddings were learned over combinations of the views in Subsection 3.1.2. When available, we also consider *GCCA-net*, where in addition to the four text views, we also include the follower and friend network views used by *NetSim-PCA*. For

computational efficiency, each of these views was first reduced in dimensionality by projecting its *BOW* TF-IDF-weighted representation to a 1000-dimensional vector through PCA.[3] We add an identity matrix scaled by a small amount of regularization, 10^{-8}, to the per-view covariance matrices before inverting, for numerical stability, and use the formulation of *GCCA* reported in Van De Velden and Bijmolt (2006), which ignores rows with missing data (some users had no data in the mention tweet view and some users accounts were private). We tune the weighting of each view i, $w_i \in \{0.0, 0.25, 1.0\}$, discriminatively for each task, although the *GCCA* objective is unsupervised once the w_i are fixed (weighting swept over only for linear *GCCA* embeddings).

When learning deep GCCA (*dGCCA*) and *LasCCA* embeddings, we do not apply any view-weighting[4]. For *LasCCA*, we consider embeddings learned over the following sets of views: {ego text, friend network}, all four text views, and all views (all text views along with two network views). We run the *LasCCA* algorithm for a fixed 100 epochs and a maximum of 20 for solving linear least squares subproblems[5]

When we compare representations in the following tasks, we sweep over embedding width in $\{10, 20, 50, 100, 200, 300, 400, 500, 1000\}$ for all methods. We also consider concatenations of vectors for every possible subset of views: singletons, pairs, triples, and all views for the *BOW-PCA* baseline.

[3] We excluded count vectors from the *GCCA* experiments for computational efficiency since they performed similarly to TF-IDF representations in initial experiments.

[4] Although it is not difficult to imagine altering the *dGCCA* and *LasCCA* objectives to per-view loss weighting.

[5]

Deep GCCA Details

We trained 40 different *dGCCA* model architectures, each with identical architectures across all text and network views, where the width of the hidden and output layers, c_1 and c_2, for each view are drawn uniformly from $[10, 1000]$, and the auxiliary representation width r is drawn uniformly from $[10, c_2]$[6]. All networks used ReLUs as activation functions, and were optimized with Adam (Kingma and Ba, 2014) for 200 epochs[7]. Networks were trained on 90% of 102,328 Twitter users, with 10% of users used as a tuning set to estimate heldout reconstruction error for early stopping. We report development and test results for the best performing model for each downstream task development set. Learning rate was set to 10^{-4} with an L1 and L2 regularization constants of 0.01 and 0.001 for all weights. This setting of regularization constants led to low reconstruction error in preliminary experiments.

3.4.2 User Engagement Prediction

The goal of user engagement prediction is to determine which topics a user will likely tweet about, using the hashtags they mention as a proxy. This task is similar to hashtag recommendation for a tweet based on its contents (Kywe et al., 2012; She and Chen, 2014; Zangerle, Gassler, and Specht, 2013). Purohit et al. (2011) presented a supervised task to predict if a hashtag would appear in a tweet using features from the user's network, previous tweets, and the tweet's content.

We selected the 400 most frequently used hashtags in messages authored by our users and which first appeared in March 2015, randomly and evenly dividing them into

[6]We chose to restrict ourselves to a single hidden layer with non-linear activation and identical architectures for each view, so as to avoid a fishing expedition. If *dGCCA* is appropriate for learning Twitter user representations, then we should be able to find a good architecture with little exploration.

[7]From preliminary experiments, we found that Adam pushed down reconstruction error more quickly than SGD with momentum, and that ReLUs were easier to optimize than sigmoid activations.

development and test sets. We held out the first 10 users who tweeted each hashtag as exemplars of users that would use the hashtag in the future. We ranked all other users by the cosine distance of their embedding to the average embedding of these 10 users. Since embeddings are learned on data pre-March 2015, the hashtags cannot impact the learned representations. Performance is measured using precision and recall at k, as well as mean reciprocal rank (MRR), where a user is marked as correct if they used the hashtag. Note that this task is different than that reported in Purohit et al. (2011), since we are making recommendations at the level of users, not tweets.

3.4.3 Friend Recommendation

The goal of friend recommendation/link prediction is to recommend/predict other accounts for a user to follow (Liben-Nowell and Kleinberg, 2007).

We selected the 500 most popular accounts – which we call celebrities – followed by our users, randomly, and evenly divided them into dev and test sets. We randomly select 10 users who follow each celebrity and rank all other users by cosine distance to the average of these 10 representations. The tweets of selected celebrities are removed during embedding training so as not to influence the learned representations. We use the same evaluation as user engagement prediction, where a user is marked as correct if they follow the given celebrity.

For both user engagement prediction and friend recommendation we z-score normalize each feature, subtracting off the mean and scaling each feature independently to have unit variance, before computing cosine similarity. We select the approach and whether to z-score normalize based on the development set performance.

3.4.4 Demographic Prediction

Our final task is to infer the demographic characteristics of a user (Al Zamal, Liu, and Ruths, 2012; Chen et al., 2015).

We use the dataset from Volkova, Coppersmith, and Van Durme (2014b) and Volkova (2015) which annotates 383 users for age (old/young), 383 for gender (male/female), and 396 political affiliation (republican/democrat), with balanced classes. Predicting each characteristic is a binary supervised prediction task. Each set is partitioned into 10 folds, with two folds held out for test, and the other eight for tuning via cross-fold validation. The provided dataset contained tweets from each user, mentioned users, friends and follower networks. It did not contain the actual social networks for these users, so we did not evaluate *NetSim*, *NetSim-PCA*, or *GCCA-net* at these prediction tasks.

Each feature for feature set was z-score normalized before being passed to a linear-kernel SVM where we swept over $10^{-4}, \ldots, 10^{4}$ for the penalty on the error term, C.

3.5 Results

3.5.1 User Engagement Prediction

Table 3.1 shows results for user engagement prediction and Figure 3.2 the precision and recall curves as a function of number of recommendations. The multiview embeddings (*GCCA*, *dGCCA*, and *LasCCA*) outperform the other baselines according to precision and recall at 1000 as well as MRR (for all multiview embeddings except *GCCA* learned over text views only). Including network views (*GCCA-net* and *GCCA-sv*) improves the performance over just considering text views. The best performing

Model	Dim	P@1	P@100	P@1000	R@1	R@100	R@1000	MRR
BOW	20000	0.03/0.045	0.021/0.014	0.009/0.005	0.001/0.002	0.075/0.053	0.241/0.157	0.006/0.006
BOW-PCA	500	0.050/0.060	0.024/0.024	0.011/0.008	0.002/0.003	0.079/0.080	0.312/0.290	0.007/0.009
NetSim	NA	0.02/0.015	0.013/0.014	0.006/0.006	0.001/0.000	0.042/0.047	0.159/0.201	0.004/0.004
NetSim-PCA	300	0.020/0.025	0.019/0.019	0.010/0.008	0.001/0.002	0.068/0.074	0.293/0.299	0.006/0.006
Word2Vec	100	0.030/0.025	0.022/0.016	0.009/0.007	0.000/0.001	0.073/ 0.057	0.254/0.217	0.005/0.004
GCCA[text]	100	0.080/0.060	0.027/0.024	0.012/0.009	0.004/0.002	0.095/0.093	0.357/0.325	0.011/0.008
GCCA-sv	500	0.090/0.060	0.030/0.026	**0.012/0.010**	0.003/0.003	0.104/0.106	0.359/0.334	0.010/0.011
GCCA-net	200	0.065/0.060	0.031/0.027	0.013/0.009	**0.003/0.004**	0.105/0.113	0.360/0.346	0.011/0.011
LasCCA[ego+friendnet]	400	0.060/0.045	0.032/0.027	**0.013/0.010**	0.003/0.002	**0.111/0.115**	0.371/0.358	0.012/0.011
LasCCA[text]	300	**0.075/0.080**	0.029/0.025	**0.012/0.010**	**0.003/0.004**	0.102/0.101	0.360/0.330	0.010/0.011
LasCCA[all]	500	0.045/0.050	**0.030/0.028**	**0.012/0.010**	0.001/0.002	0.108/0.114	0.378/0.368	0.009/0.010
dGCCA[all]	710	**0.070/0.080**	**0.030/0.028**	**0.013/0.010**	**0.003/0.004**	0.105/0.112	**0.385/0.373**	**0.010/0.012**
NetSize	NA	0.000/0.000	0.001/0.000	0.001/0.001	0.000/0.000	0.001/0.002	0.012/0.012	0.000/0.000
Random	NA	0.000/0.000	0.001/0.000	0.000/0.000	0.000/0.000	0.001/0.000	0.002/0.008	0.000/0.000

Table 3.1: Macro performance at user engagement prediction on dev/test. Ranking of model performance was consistent across metrics. Precision is low since few users tweet a given hashtag. Values are bolded by best test performance according to each metric. Simple reference ranking techniques (bottom): *NetSize*: a ranking of users by the size of their local network; *Random* randomly ranks users. The *Dim* column is the dimensionality of the selected embedding.

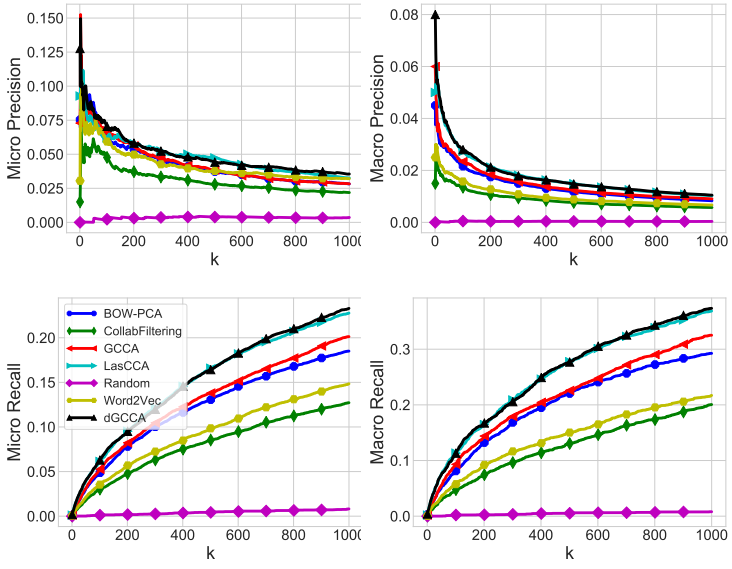


Figure 3.2: The best performing approaches on user engagement prediction as a function of number of recommendations. The ordering of methods is consistent across k. The plotted *LasCCA* is learned over all views (*[all]*).

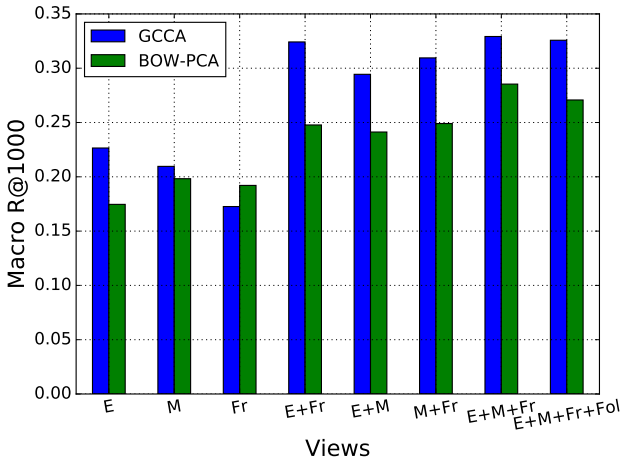


Figure 3.3: Macro recall@1000 on user engagement prediction for different combinations of text views. Each bar shows the best performing model swept over dimensionality. *E*: ego, *M*: mention, *Fr*: friend, *Fol*: follower tweet views.

GCCA setting placed weight 1 on the ego tweet view, mention view, and friend view, while *BOW-PCA* concatenated these views, suggesting that these were the three most important views but that *GCCA* was able to learn a better representation. Figure 3.3 compares performance of different view subsets for *GCCA* and *BOW-PCA*, showing that *GCCA* uses information from multiple views more effectively for predicting user engagement.

There are a few other several points to note: First is that *dGCCA* outperforms linear multiview methods according to recall at 1000 and MRR. This is exciting because this task benefits from incorporating more than just two views from Twitter users through linear multiview representation learning methods. These results suggest that a nonlinear transformation of the input views can yield additional gains in performance. In addition, *GCCA* models sweep over every possible weighting of views with weights in $\{0, 0.25, 1.0\}$. *GCCA* has a distinct advantage in that the model is allowed to discriminatively weight views to maximize downstream performance. The fact that

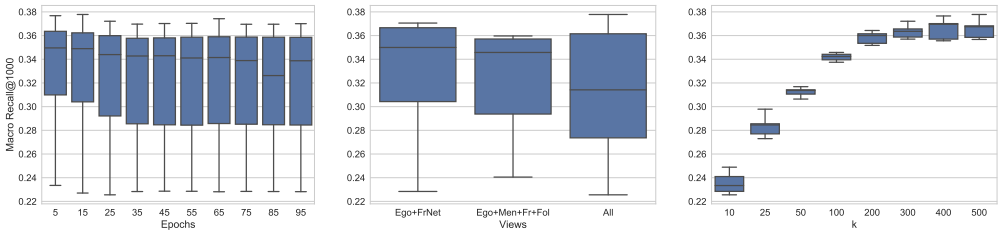


Figure 3.4: Development macro recall at 1000 recommendations for *LasCCA* embeddings at the user engagement task. Boxplots collapse performance across a full sweep of *(left)* number of *LasCCA* epochs, *(center)* view subsets, and *(right)* embedding width.

dGCCA is able to outperform *GCCA* at hashtag recommendation is encouraging, since *GCCA* has much more freedom to discard uninformative views, whereas the *dGCCA* objective forces networks to minimize reconstruction error equally across all views.

In addition, the *LasCCA* embeddings learned on all views, also unweighted, perform almost as well as *dGCCA*. This suggests that linear multiview representation learning methods may learn similarly effective embeddings as nonlinear ones, only with a slightly different *GCCA* formulation. However, it is not clear why the SUMCOR objective would be more appropriate than the MAXVAR generalized CCA objective for learning embeddings geared towards this task. It also further underscores the fact that multiview techniques seem to be more appropriate than single-view for learning embeddings effective at user engagement prediction.

Effect of *LasCCA* Solution Quality on User Engagement Prediction

LasCCA is an iterative algorithm for solving the SUMCOR-GCCA problem. In practice we learned embeddings with a fixed 100 epochs for all embedding widths. How many times must we iterate to achieve good downstream performance? To test this, we varied the number of epochs training each *LasCCA* embedding by

Model	Dim	P@1	P@100	P@1000	R@1	R@100	R@1000	MRR
BOW	20000	0.164/0.268	0.164/0.232	0.133/0.153	0.000/0.000	0.005/0.007	0.043/0.048	0.000/0.001
BOW-PCA	20	0.480/0.500	0.415/0.421	0.311/0.314	0.000/0.000	0.014/0.014	0.101/0.102	0.001/0.001
NetSim	NA	0.672/0.680	0.575/0.582	0.406/0.420	0.000/0.000	0.019/0.019	0.131/0.132	**0.002/0.002**
NetSim-PCA	500	**0.720/0.736**	**0.596/0.601**	**0.445/0.439**	0.000/0.000	**0.020/0.020**	**0.149/0.147**	**0.002/0.002**
Word2Vec	200	0.436/0.340	0.344/0.320	0.260/0.249	0.000/0.000	0.011/0.010	0.084/0.080	0.001/0.001
GCCA	50	0.484/0.472	0.370/0.381	0.269/0.276	0.000/0.000	0.012/0.013	0.089/0.091	0.001/0.001
GCCA-sv	500	**0.720/0.736**	**0.596/0.601**	**0.445/0.439**	0.000/0.000	**0.020/0.020**	**0.149/0.147**	**0.002/0.002**
GCCA-net	20	0.520/0.544	0.481/0.475	0.376/0.364	0.000/0.000	0.016/0.016	0.123/0.120	0.001/0.001
LasCCA[ego+friendnet]	50	0.352/0.326	0.328/0.302	0.235/0.239	**0.000/0.001**	0.010/0.010	0.077/0.078	0.001/0.001
LasCCA[text]	50	0.352/0.412	0.223/0.318	0.244/0.250	0.000/0.000	0.010/0.010	0.079/0.080	0.001/0.001
LasCCA[all]	50	0.420/0.448	0.335/0.342	0.260/0.263	0.000/0.000	0.011/0.011	0.085/0.086	0.001/0.001
dGCCA[all]	185	0.512/0.544	0.400/0.411	0.297/0.302	0.000/0.000	0.013/0.014	0.099/0.100	0.001/0.001
NetSize	NA	0.180/0.176	0.025/0.026	0.033/0.035	0.000/0.000	0.001/0.001	0.009/0.010	0.000/0.000
Random	NA	0.072/0.136	0.040/0.041	0.034/0.036	0.000/0.000	0.001/0.001	0.010/0.010	0.000/0.000

Table 3.2: Macro performance for friend recommendation. Performance of *NetSim-PCA* and *GCCA-sv* are identical since the view weighting for *GCCA-sv* only selected solely the friend view. Thus, these methods learned identical user embeddings.

increments of 5 up to 100. We then examined the final downstream performance at user engagement prediction as a function of how many *LasCCA* epochs were taken to learn an embedding as well as other training parameters such as embedding width and which views to apply *LasCCA* too.

It is encouraging that performance at hashtag recommendation is completely insensitive to number of epochs (Figure 3.4, left). In contrast, downstream performance is most influenced by which embedding width we choose (Figure 3.4, right), although we also find that *LasCCA* user embeddings learned over network views improve over just text views (center), echoing what we see when learning *GCCA* embeddings.

3.5.2 Friend Recommendation

Table 3.2 shows results for friend prediction and Figure 3.5 similarly shows that performance differences between approaches are consistent across k (number of recommendations.) Adding network views to *GCCA*, *GCCA-net*, improves performance, although it cannot contend with *NetSim* or *NetSim-PCA*, although *GCCA-sv* is able to meet the performance of *NetSim-PCA*. The best *GCCA* placed non-zero weight on the friend tweets view, and *GCCA-net* only places weight on the friend network view;

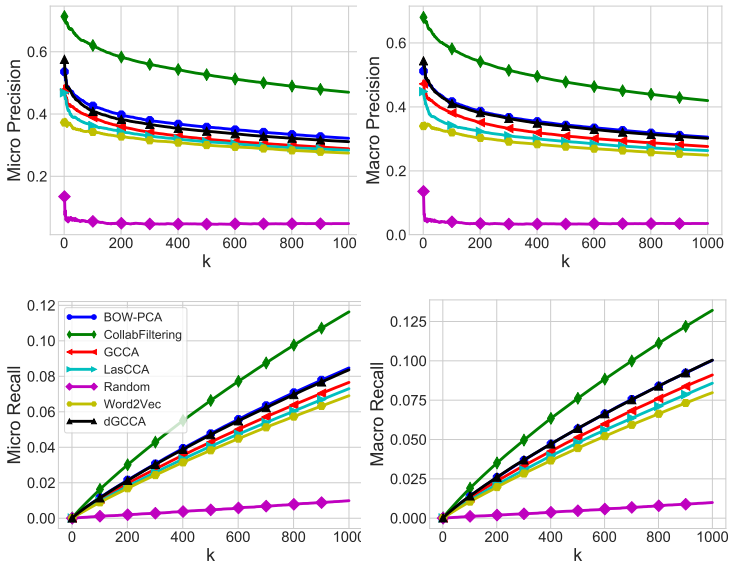


Figure 3.5: Performance of user embeddings at friend recommendation as a function of number of recommendations.

the other views were not informative. *BOW-PCA* and *Word2Vec* only used the friend tweet view. This suggests that the friend view is the most important for this task, and multiview techniques cannot exploit additional views to improve performance. *GCCA-sv* performs identically to *GCCA-net* since it only placed weight on the friend network view, learning identical embeddings to *GCCA-net*.

Since only the friend network view was useful for learning representations for friend recommendation, it is unsurprising that *dGCCA* when applied to all views cannot compete with *GCCA* representations learned on the single useful friend network view[8]. The same holds for embeddings learned by *LasCCA* over several unweighted views.

[8]The performance of WGCCA suffers compared to PCA because whitening the friend network data ignores the fact that the spectrum of the decays quickly with a long tail – the first few principal components made up a large portion of the variance in the data, but it was also important to compare users based on other components.

Model	age	gender	politics
BOW	0.771/0.740	0.723/0.662	0.934/0.975
BOW-PCA	0.784/0.649	0.719/0.662	0.908/0.900
BOW-PCA + BOW	0.767/0.688	0.660/0.714	**0.937/0.9875**
Word2Vec	**0.790/0.753**	**0.777/0.766**	0.927/0.938
GCCA	0.725/0.740	0.742/0.714	0.899/0.8125
GCCA + BOW	0.764/0.727	0.657/0.701	0.940/0.9625
GCCA-sv	0.709/0.636	0.699/0.714	0.871/0.850
GCCA-sv + BOW	0.761/0.688	0.647/0.675	0.937/0.9625
LasCCA[text]	0.689/0.662	0.712/0.662	0.883/0.838
LasCCA[text] + *BOW*	0.754/0.662	0.666/0.649	0.931/ 0.950
dGCCA	0.735/0.727	0.699/0.649	0.845/0.800
dGCCA + BOW	0.771/0.649	0.673/0.610	0.931/0.950

Table 3.3: Average CV/test accuracy for inferring demographic characteristics given different feature sets.

3.5.3 Demographic Prediction

Table 3.3 shows the average cross-fold validation and test accuracy on the demographic prediction task. + *BOW* indicates that *BOW* features were concatenated to the embeddings as an additional feature set for the classifier. The wide variation in performance is due to the small size of the datasets, thus it's hard to draw many conclusions (the average development performance of all models are within one standard deviation of each other). However, *Word2Vec* surpasses other representations in two out of three datasets, and including a TF-IDF weighted bag of words features tends to improve the generalization performance of most classifiers.

It is difficult to compare the performance of the methods we evaluate here to that reported in previous work (Al Zamal, Liu, and Ruths, 2012). This is because they report cross-fold validation accuracy (not test), they consider a wider range of hand-engineered features, different subsets of networks, radial basis function kernels

for SVM, and find that accuracy varies wildly across different feature sets. They report cross-fold validation accuracy ranging from 0.619 to 0.805 for predicting age, 0.560 to 0.802 for gender, and 0.725 to 0.932 for politics.

3.5.4 Evaluating User Cluster Coherence

Although we evaluated embeddings on several quantitative tasks, these experiments do not tell us which embedding type best captures intuitive notions of user groups, similar to how word embeddings have been shown to cluster words with similar meaning or syntactic properties together in embedding space. In order to evaluate how well different embeddings captured human notions of types of people or user groups, we performed the following experiment.

We considered three different types of 500-dimensional user embeddings: *BOW-PCA* only on ego text (*BOW-PCA[ego]*), *BOW-PCA* on the concatenation of all views (*BOW-PCA[all]*), and *dGCCA* on all views[9]. For each embedding type, we fit a 50-cluster Gaussian mixture model with diagonal covariance matrix, for 100 expectation maximization iterations. For all users in our data, we assigned them to the most probable Gaussian according to the mixture model, and we selected the five most probable users assigned to that cluster under the Gaussian distribution. We ensured that each of these accounts were still active by querying their user summary page (`https://twitter.com/intent/user?user_id=${USER_ID}`) and excluded the cluster from further analysis if we could not find five active users within the closest 10 users to the Gaussian centroid. This yielded a total of 144 clusters: 50 clusters for *BOW-PCA[ego]* and *BOW-PCA[all]* each, and 44 clusters for *dGCCA*.

[9]We consider *dGCCA* to represent multiview embeddings in general, because it was the best performing multiview method at user engagement prediction.

3.5.4.1 Experiment

We used these cluster exemplars to construct an intruder detection task to submit to Amazon Mechanical Turk[10]. For each cluster, we presented the subject with links to four of the five exemplar users' Twitter summaries[11], along with an intruder user sampled uniformly at random from another cluster's exemplars. The order of users was randomized for each HIT and the subject was asked to complete two tasks:

1. Given only the information provided on the users' summary pages (their most recent tweets, user text description, and profile image), identify which user is the most different from the other four.

2. Describe in your own words, and as specifically as possible, what the other four users have in common.

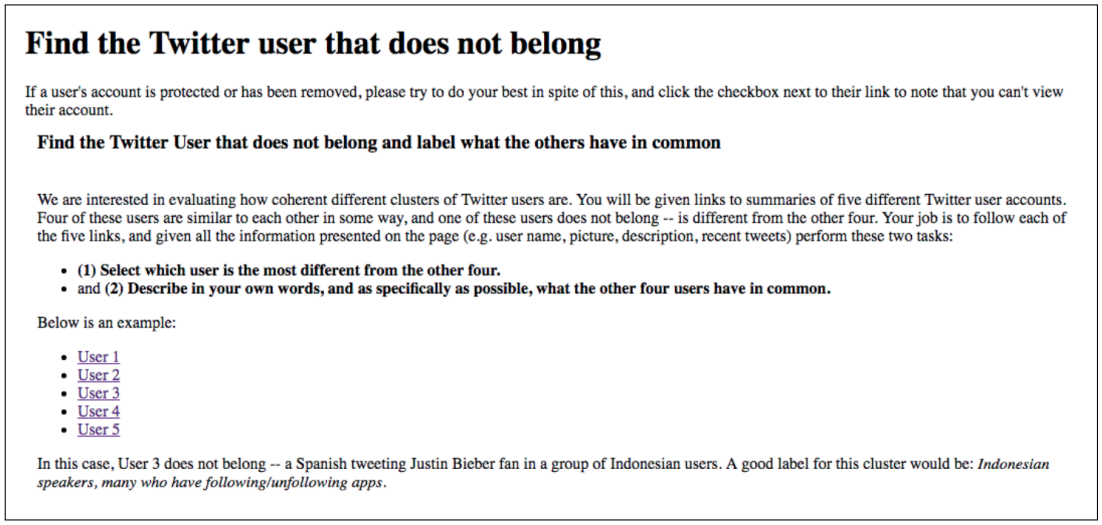
Screenshots of the Mechanical Turk instructions and a sample HIT are presented in Figures 3.6 and 3.7.

We treat the intruder detection task as a proxy for user cluster coherence – how similar are users belonging to the same cluster. If it is easier for a subject to spot which user does not belong, that suggests that the other users share an easily identifiable, common property. This task was inspired by work in evaluating the quality of topics learned by a topic model, specifically the *word intrusion* task described in Chang et al. (2009). In addition, we were able to use this task to quickly collect cluster labels for qualitative analysis, without influence from our own biases.

Each cluster was labelled by three unique subjects and we compare embedding types by accuracy at the intruder detection task, averaged over all annotations.

[10]This experiment was submitted in July 2018, over three years after the data used to learn embeddings were collected.

[11]`https://twitter.com/intent/user?user_id=[USER_ID]`



Find the Twitter user that does not belong

If a user's account is protected or has been removed, please try to do your best in spite of this, and click the checkbox next to their link to note that you can't view their account.

Find the Twitter User that does not belong and label what the others have in common

We are interested in evaluating how coherent different clusters of Twitter users are. You will be given links to summaries of five different Twitter user accounts. Four of these users are similar to each other in some way, and one of these users does not belong -- is different from the other four. Your job is to follow each of the five links, and given all the information presented on the page (e.g. user name, picture, description, recent tweets) perform these two tasks:

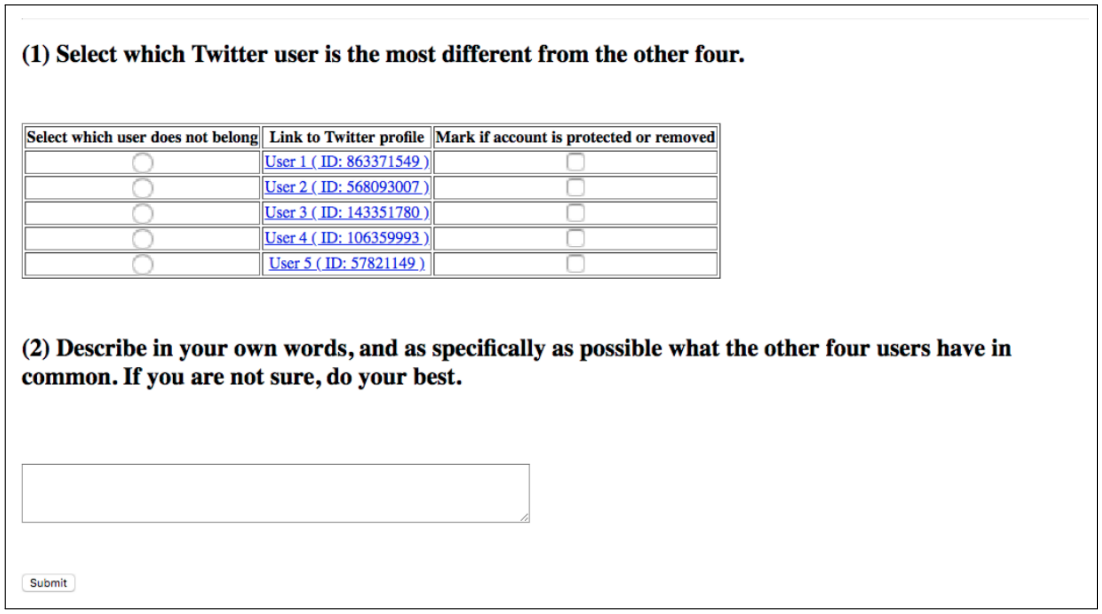
- **(1) Select which user is the most different from the other four.**
- and **(2) Describe in your own words, and as specifically as possible, what the other four users have in common.**

Below is an example:

- User 1
- User 2
- User 3
- User 4
- User 5

In this case, User 3 does not belong -- a Spanish tweeting Justin Bieber fan in a group of Indonesian users. A good label for this cluster would be: *Indonesian speakers, many who have following/unfollowing apps*.

Figure 3.6: Mechanical Turk instructions for the user cluster intruder detection task.



(1) Select which Twitter user is the most different from the other four.

Select which user does not belong	Link to Twitter profile	Mark if account is protected or removed
○	User 1 (ID: 863371549)	☐
○	User 2 (ID: 568093007)	☐
○	User 3 (ID: 143351780)	☐
○	User 4 (ID: 106359993)	☐
○	User 5 (ID: 57821149)	☐

(2) Describe in your own words, and as specifically as possible what the other four users have in common. If you are not sure, do your best.

Submit

Figure 3.7: Example assignment for the user cluster intruder detection HIT. The user ID links point the subject to a Twitter user's summary page.

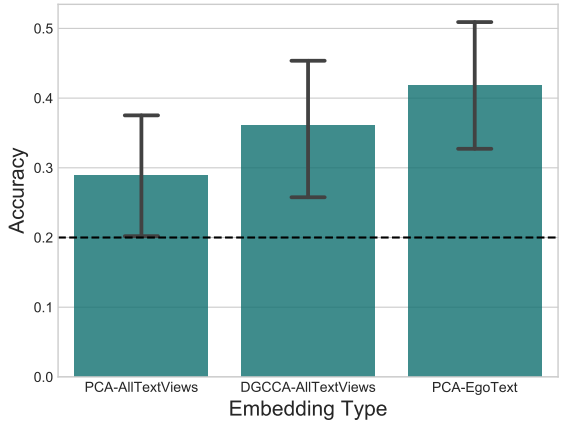


Figure 3.8: Average Turker accuracy at selecting the intruder out of a cluster of five users. The horizontal line marks performance of random guessing (20%). 95% confidence interval bars are generated by 10,000 bootstrap samples with resampling of the same size as the original sample. Each bar corresponds to a different type of embedding from which clusters were induced.

3.5.4.2 Results

We omitted a single subject's HITs from analysis as they completed a large number of HITs very quickly, performed only slightly above random chance (23% accuracy), and labelled clusters uninformatively (e.g. "They are all the same" or "'posts are in English"). After removing this user, we calculated accuracy over a total of 311 annotations (110 for *BOW-PCA[ego]*, 104 for *BOW-PCA[all]*, and 97 for *dGCCA*). Surprisingly, subjects found the clusters from *BOW-PCA[ego]* tended to be the most coherent (Figure 3.8).

Although the confidence intervals estimated by bootstrap samples are wide, subjects were able to detect the intruder statistically significantly more frequently than chance for all embedding types according to a proportion z-test ($p = 0.05$)[12]. The

[12]From the `statsmodels` python library:
`statsmodels.stats.proportion.proportions_ztest`

BOW-PCA[ego] embeddings resulted in statistically significantly higher accuracy than PCA on all views according to this same test ($p = 0.05$).

One reason why subjects better detected intruders in the *BOW-PCA[ego]* clusters was likely because of the information they were allowed to act on: a short summary of the Twitter user. Grouping users together by the frequent words they post is a simple cue for someone to latch onto. These are exactly the sorts of features that methods that only consider the ego text view will try to preserve. However, we fidn it interesting that *dGCCA* clusters are more coherent than *BOW-PCA*[all]. Although less coherent than an ego text embedding, this suggests that multiview representation learning methods yield more "natural" user embeddings when consolidating multiple types of input behavior.

Appendix A contains an exhaustive list of labels assigned to each cluster along with a few examples of Twitter users belonging to the same cluster. Many of these clusters were assigned vague labels ("They all speak English", "none"), which speaks to the difficulty of this task. Not only are the user clusters noisy and subjects are given scant information in the Twitter summary, but the user embeddings were learned *over three years before* the HIT was conducted.

Preprocessing Considerations In this chapter we naïvely preprocessed the text views by removing stop words and restricting the vocabulary size to the 20,000 most frequent token types. Because of this, the user representations we learn in this chapter sometimes captured user behavior that would be considered noise in most downstream tasks.

Figure 3.9: Tweets from exemplar users from an "astrology app" cluster. Members of this cluster belonged to a range of astrological signs and the only discernible feature shared between them were automated posts generated by the app. We intentionally obfuscated the users' names for their privacy.

The most salient examples of this were clusters of users who registered for the same Twitter app. Astrological sign apps are particularly popular, and some automatically post tweets associated with the user's astrological sign. Figure 3.9 shows exemplar tweets from one such cluster learned from *GCCA* embeddings. This cluster mixes users with different signs suggesting that user representations generalize to those who subscribed to this particular astrology app, rather than homing in on repetition of tokens for one astrological sign. Another cluster included users who subscribed to follower-tracking apps that automatically tweet about changes in their follower network. Although we focus on evaluating different *methods* of learning user representations, this underscores just how important data quality and preprocessing are when applying these methods to real-world data.

3.6 Summary

This chapter shows how unsupervised user embeddings can be learned from multiple views of Twitter user behavior. We find that although embeddings learned on friending behavior alone are the most predictive of other friends a user may have, multiview embeddings learned over views of both what the ego user posts and their friending behavior better capture which hashtags they are likely to use in the future. Although subjects found embeddings learned only on ego text to yield more coherent user clusters than multiview embeddings, multiview user embedding clusters were more coherent than those learned by applying a single-view dimensionality reduction technique to all views.

Chapter 4

User-Conditioned Topic Models

Chapter 3 described different unsupervised methods for learning social media user embeddings, and primarily evaluated these embeddings intrinsically – by how well they capture similar topic posting or friending behavior. In this chapter we present a non-traditional application of user features, showing how they can be used to improve topic modeling of social media text.

This chapter highlights the breadth of applications that can benefit from user features. Latent Dirichlet Allocation (*LDA*) is the most common topic model applied by social scientists to uncover themes in large corpora. We show that opting to fit a supervised topic model with user features as supervision can lead to improved model fit and better guide the topics that are learned. We also present a new topic model, deep Dirichlet Multinomial Regression (*dDMR*), that can better make use of highy-dimensional and only distantly related features, improving upon a previous supervised topic model, Dirichlet Multinomial Regression (*DMR*).

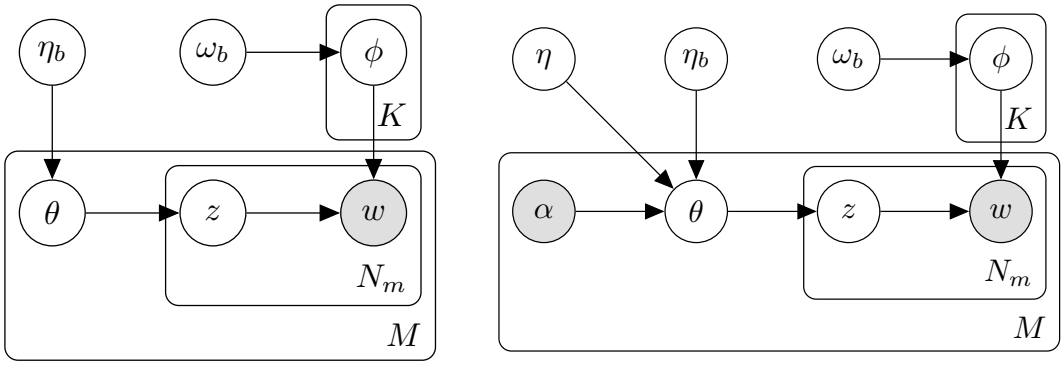
Section 4.1 gives background on upstream supervised topic models: their basic generative story, how one fits these models by collapsing Gibbs sampling, and how they compare to unsupservised topic models. Section 4.2 describes *dDMR* and

analyzes which types of corpora it excels at modeling by synthetic data experiments. Section 4.3 evaluates *dDMR* against other unsupervised and supervised models on three datasets: a collection of New York Times articles, Amazon product reviews, and Reddit messages. Section 4.4 finally applies *dDMR* to modeling three public policy-related Twitter datasets using features derived from inferred user location as supervision. The Twitter distant user feature supervision experiments were published as Benton et al. (2016b), a long paper in AAAI 2016, and also appear in chapter 6 of Michael J. Paul's Ph.D. thesis (Paul, 2015a). The *dDMR* model definition and evaluations were published as a long paper in NAACL 2018 (Benton and Dredze, 2018a).

4.1 Background: Supervised Topic Models

Social media has proved invaluable for research in social and health sciences, including sociolinguistics (Eisenstein, Smith, and Xing, 2011), political science (O'Connor et al., 2010b), and public health (Paul and Dredze, 2011). A common theme is the use of topic models (Blei, Ng, and Jordan, 2003), which, by identifying major themes in a corpus, summarize the content of large text collections. Topic models have been applied to characterize tweets (Ramage, Dumais, and Liebling, 2010), blog posts and comments (Yano, Cohen, and Smith, 2009; Paul and Girju, 2009), and other short texts (Phan, Nguyen, and Horiguchi, 2008).

Latent Dirichlet Allocation (*LDA*) is a fully unsupervised generative model, which may have limited utility when trying to learn topics that capture the opinions of document authors. Supervised topic models offer one option for guiding topics and improving model fit. Supervised topic models come in many flavors, such



(a) ***LDA*** (b) ***DMR***

Figure 4.1: Graphical model of *LDA* (left) and *DMR* (right) in plate notation. The key difference between these topic models is that *DMR* includes document-dependent features, α, that affect the document-topic prior through log-linear weights, η, shared across all documents. *LDA* conversely shares the same document-topic prior for all documents.

as predicting labels for each document, e.g., supervised LDA (Mcauliffe and Blei, 2008); modeling tags associated with each document, e.g., labeled LDA (Ramage et al., 2009) or tagLDA (Zhu, Blei, and Lafferty, 2006); placing priors over topic-word distributions (Jagarlamudi, III, and Udupa, 2012; Paul and Dredze, 2013); or interactive feedback from the user (Hu et al., 2014). Using the terminology of Mimno and McCallum (2008), these models can be classified as either "Upstream" or "Downstream", referring to whether this supervision is assumed to be generated before or after the text in the generative stories. The supervised models we consider in this chapter are upstream models with document-level supervision, in particular Dirichlet Multinomial Regression (*DMR*).

4.1.1 *DMR* Generative Story

4.1 illustrates the generative story of *LDA* and *DMR* in plate notation. In upstream topic models, supervision influences the *priors* over topic distributions in documents.

1. For each document m:
   (a) $\tilde{\theta}_{mk} \leftarrow \exp(\eta_{bk})$, for each topic k
   (b) $\tilde{\theta}_{mk} \leftarrow \tilde{\theta}_{mk} * \exp(\alpha_m^T \eta_k)$, for each topic k
   (c) $\theta_m \sim \text{Dirichlet}(\tilde{\theta}_m)$
2. For each topic k:
   (a) $\tilde{\phi}_{kv} = \exp(\omega_{bv})$
   (b) $\phi_k \sim \text{Dirichlet}(\tilde{\phi}_k)$
3. For each token n in each document m:
   (a) Sample topic index $z_{mn} \sim \theta_m$
   (b) Sample word token $w_{mn} \sim \phi_{z_{mn}}$

Figure 4.2: Generative story for *DMR*. Differences between *LDA* and *DMR* are written in red.

In *DMR* this is done by parameterizing the document-topic Dirichlet prior as a log-linear function of the document labels $\boldsymbol{\alpha}$ and regression coefficients $\boldsymbol{\eta}$. Under a *DMR* topic model (the basic upstream model in our experiments), each document has its own Dirichlet($\tilde{\theta}_m$) prior, with $\tilde{\theta}_{mk}$=$\exp(\eta_{bk} + \alpha_m^T \eta_k)$, where α_m is the supervision feature vector of the m^{th} document, η_k is the k^{th} topic's feature coefficients, and η_{bk} is a bias term for topic k (intercept). For positive $\eta_k^{(i)}$, the prior for topic k in document m will increase as $\alpha_m^{(i)}$ increases, while negative $\eta_k^{(i)}$ will decrease the prior weight.

Figure 4.2 provides the generative story for *DMR*, with the portions that differ from *LDA* in red.

4.1.2 Fitting Topic Models

The experiments described in this chapter use a collapsed Gibbs sampler with regularized hyperparameter updates to infer topic model parameters. Methods such as variational expectation maximization are also possible, but we only fit models by

Gibbs sampling updates due to its simplicity of implementation and applicability to all the topic model architectures we consider.

The inference procedure for each model involves alternating between one iteration of collapsed Gibbs sampling (sampling each token's topic assignment) and one iteration of gradient ascent for the parameters η_b (bias vector in the document-topic prior), ω_b (bias in the topic-word prior), and η (weights determining how document supervision influences the document-topic prior).

Gibbs Sampling

The Gibbs sampling step involves sampling z_{mn}, each topic assignment, in turn for every word in the corpus, w_{mn}, where m is the document index and n is the word index within a document. Each topic assignment is drawn conditioned on all previous topic assignments as well as the document-topic and topic-word priors, $Dirichlet(\widetilde{\theta}_m)$ and $Dirichlet(\widetilde{\phi})$ respectively. Formally, the probability that k is sampled as the current topic assignment is proportional to:

$$\begin{aligned} &p(z_{mn} = k | \{z \text{ s.t. } z \neq z_{mn}\}, \widetilde{\theta}_m, \widetilde{\phi}) \propto \\ &(C(m,k) + \widetilde{\theta}_{mk})(\frac{C(w_{mn},k) + \widetilde{\phi}_{(w_{mn})}}{\sum_v C(v,k) + \widetilde{\phi}_v}) \end{aligned} \tag{4.1}$$

where $C(m,k)$ is the number of times topic k was sampled in document m (excluding the word we are currently sampling for) and $C(w_{mn},k)$ is the number of times topic k was sampled for word w_{mn} (Paul, 2015b). The counts are aggregated over all topic assignments except the current word being sampled. The term on the right-hand side can be converted to a probability by normalizing by the sum of unnormalized topic sampling probabilities:

$$Z = \sum_{k=1}^{K} p(z_{mn} = k | \{z \text{ s.t. } z \neq z_{mn}\}, \widetilde{\theta}_m, \widetilde{\phi}) \tag{4.2}$$

This Gibbs sampling step is the same for both unsupervised *LDA* as well as supervised models like *DMR* – the only difference between these two models is how $\widetilde{\phi}$ and $\widetilde{\theta}$ are parameterized (Figure 4.2).

Hyperparameter Updates

The hyperparameters that parameterize the document-topic prior are learned by first-order methods. We first calculate the gradient of the joint log-likelihood of both the observed words and sampled topics with respect to the prior hyperparameters, and then update the hyperparameters along a descent direction[1].

The partial derivative of an upstream topic model's log-likelihood with respect to the document-topic prior $\widetilde{\theta}_m$:

$$\begin{aligned} \frac{\delta \log \mathcal{L}(z|\widetilde{\theta}_m)}{\delta \widetilde{\theta}_{mk}} =& \psi(C(m,k) + \widetilde{\theta}_{mk}) - \psi(\widetilde{\theta}_{mk}) + \\ & \psi(\sum_{k'=1}^{K} \widetilde{\theta}_{mk}) - \psi(\sum_{k'=1}^{K} C(m,k') + \widetilde{\theta}_{mk}) \end{aligned} \tag{4.3}$$

[1] In practice we do not use a fixed learning rate, choose the exact gradient as a descent direction, but instead use adaptive learning rate methods to update the prior hyperparameters: AdaDelta (Zeiler, 2012) or AdaGrad (Duchi, Hazan, and Singer, 2011).

where k is a topic index and ψ is the *digamma* function, the derivative of the natural logarithm of the gamma function (a generalization of factorial to complex and real numbers). The partial derivative with respect to $\widetilde{\phi}$ is:

$$\frac{\delta \mathcal{L}(w|z,\widetilde{\phi})}{\delta \widetilde{\phi}_w} = \sum_{k=1}^{K} \psi(C(w,k) + \widetilde{\phi}_w) - \psi(\widetilde{\phi}_w) + \psi(\sum_{w=1}^{W} \widetilde{\phi}_{w'}) - \psi(\sum_{w=1}^{W} C(w',k) + \widetilde{\phi}_{w'}) \tag{4.4}$$

where w and w' are word indices. If $\widetilde{\theta}_m$ is parameterized as $exp(\eta_b + \eta^T \alpha_m)$ (as in *DMR*), we can simply apply the chain rule to solve for the partial derivative with respect to the prior hyperparameters η_b and η. The same goes for the topic-word parameters ω_b.

In practice we also include a small amount of ℓ_2 regularization on the gradient term. This is necessary to prevent hyperparameter weights from growing far too large, overfitting to the current topic samples.

Calculating Model Fit

Supervised classifiers are typically evaluated according to predictive performance on some heldout data, unobserved during training. Topic models (and unsupervised models in general) are trickier to evaluate since the quality of a topic model is ultimately determined by how coherent or interpretable the learned topics are according to the human reviewing them. Needless to say, human-judged interpretability is not a scalable measure of model quality. It cannot be easily used for model selection: how many topics should my model learn; what kind of document supervision should I condition the model on? It also cannot be used to decide when the optimization algorithm has converged to a good solution. Instead, we use heldout perplexity in

many of our experiments to decide when a model has converged and which model to select.

Perplexity is just the exponentiated average negative log probability of the corpus under the model:

$$Perplexity(w|z,\widetilde{\theta},\widetilde{\phi}) = exp\left(\frac{-\sum_{m=1}^{M}\sum_{n=1}^{N_m}\log p(w_{mn}|z_{mn},\widetilde{\theta}_m,\widetilde{\phi})}{\sum_{m=1}^{M}N_m}\right) \tag{4.5}$$

where N_m is the number of words in document m. Perplexity can be interpreted as encoding how "confused" the topic model is on average for each token in the corpus. A topic model with lower perplexity is better at predicting which words are likely to occur in a document than one with higher perplexity (assigning higher average log-likelihood to words in the corpus).

Heldout perplexity is computed by only aggregating document-topic and topic-word counts from every *other* token in the corpus, and evaluating perplexity on the remaining heldout tokens. This corresponds to the "document completion" evaluation method as described in (Wallach et al., 2009), where instead of holding out the words in the second half of a document, every other token is held out after shuffling the words within a document[2]. The counts $C(m,k)$, $C(w,k)$ are computed only over training token samples.

[2]Word ordering within a document is of no consequence to the probabilistic topic models we consider, since they assume that each word is generated independently of all other words in a document (given the current document-topic distribution). We shuffle the tokens within each document before topic sampling to ensure that ordering effects do not influence the word distributions between training and heldout tokens.

4.2 Deep Dirichlet Multinomial Regression (*dDMR*)

Problems with *DMR* Document collections are often accompanied by metadata and annotations, such as a book's author, an article's topic descriptor tags, images associated with a product review, or structured patient information associated with clinical records. These document-level annotations provide additional supervision for guiding topic model learning. *DMR* is an upstream topic model with a particularly attractive method for incorporating arbitrary document features. Rather than defining specific random variables in the graphical model for each new document feature, *DMR* treats the document annotations as features in a log-linear model. By making no assumptions on model structure of new random variables, *DMR* is flexible to incorporating different types of features.

Despite this flexibility, *DMR* models are typically restricted to a small number of document features. Several reasons account for this restriction: (1) Many text corpora only have a small number of document-level features; (2) Model hyperparameters become less interpretable as the dimensionality grows; and (3) *DMR* is liable to overfit the hyperparameters when the dimensionality of document features is high. In practice, applications of *DMR* are limited to settings with a small number of features, or where the analyst selects a few meaningful features by hand.

Proposal: Neuralize the Prior One solution to addressing this restriction is to learn low dimensional representations of document metadata before conditioning *DMR* on them. Neural networks have shown wide-spread success at learning generalizable representations, often obviating the need for hand designed features (Collobert and Weston, 2008). A prime example is word embedding features in natural language

processing, which supplant traditional lexical features (Brown et al., 1992; Mikolov et al., 2013a; Pennington, Socher, and Manning, 2014). Jointly learning networks that construct feature representations along with the parameters of a standard NLP model has become a common approach. For example, Yu, Gormley, and Dredze (2015) used a tensor decomposition to jointly learn features from both word embeddings and traditional NLP features, along with the parameters of a relation extraction model. Additionally, neural networks can handle a variety of data types including text, images, and general metadata features. This makes them appropriate tools for addressing dimensionality reduction in *DMR*.

Deep Dirichlet Multinomial Regression (*dDMR*) is a model that extends *DMR* by introducing a deep neural network that learns a transformation of the input metadata into features used to form the document-topic prior. Whereas *DMR* parameterizes the document-topic priors as a log-linear function of document features, *dDMR* jointly learns a feature representation for each document along with a log-linear function that best captures the distribution over topics. Since the function mapping document features to topic prior is a neural network, we can jointly optimize the topic model and the neural network parameters by gradient ascent and back-propagation.

4.2.1 Model

dDMR extends *DMR* by replacing the document supervision (vector), α, in the document-topic Dirichlet prior with a supervision embedding learned by a function f mapping arbitrary document supervision to a real-valued vector, $\alpha' = f(\alpha)$. For simplicity we make no assumptions on the type of this function, only that it can be optimized to minimize a cost on its output by gradient ascent. In practice, we define this function as a neural network, where the architecture of this network is informed

by the type of document supervision, e.g. a convolutional neural network for images. We use neural networks since they are expressive, generalize well to unseen data, and can be jointly trained using straightforward gradient ascent with back-propagation.

The generative story for *dDMR* is as follows:

1. Representation function $f \in \mathbb{R}^F \rightarrow \mathbb{R}^K$
2. Topic-word prior parameters: $\omega_b \in \mathbb{R}^V$
3. For each document m with features $\alpha_m \in \mathbb{R}^F$, generate document prior:
   (a) $\widetilde{\theta}_m = exp(f(\alpha_m))$
   (b) $\theta_m \sim Dirichlet(\widetilde{\theta}_m)$
4. For each topic k, generate word distribution:
   (a) $\phi_k \sim Dirichlet(exp(\omega_b)$
5. For each token generate corpus:
   (a) Topic (unobserved): $z_{mn} \sim \theta_m$
   (b) Word (observed): $w_{mn} \sim \phi_{z_{mn}}$

where V is the vocabulary size and K are the number of topics. In practice, the document features need not be restricted to fixed-length feature vectors, e.g. f may be an RNN that maps from a sequence of characters to a fixed length vector in $\mathbb{R}^k$. *DMR* is a special case of *dDMR* with the choice of a linear function for f. Figure 4.3 displays the graphical model diagram for *dDMR*.

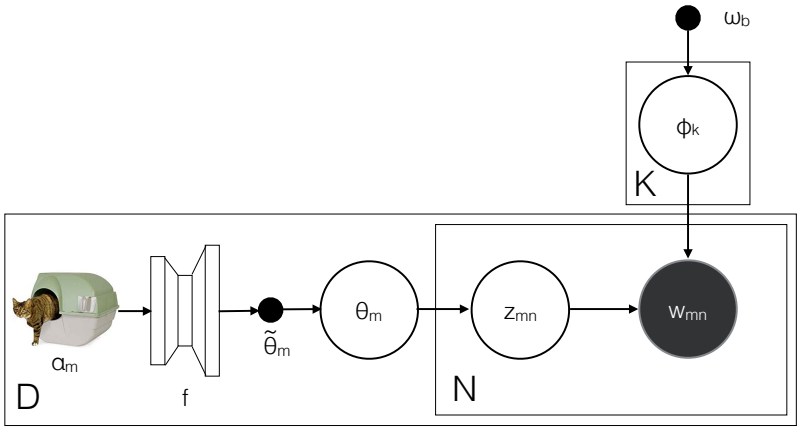


Figure 4.3: Plate diagram of *dDMR*. f is depicted as a feedforward fully-connected network, and the document features are given by an image – in this case a picture of a cat.

4.2.2 Synthetic Experiments

Our intuition in developing *dDMR* was that if the document-level supervision is very high-dimensional but lies on a low-dimensional manifold, then expressing the supervision with respect to its position on this manifold will avoid overfitting a topic model to the training corpus. *DMR* does not perform any such dimensionality reduction and thus may be liable to overfitting when the source of supervision is high-dimensional; a neural prior topic model that learns an appropriate embedding of the supervision will not be as susceptible. We constructed a synthetic dataset to determine what sort of corpora are more appropriate to model with *dDMR* rather than *DMR*.

Data Generation

Algorithm 3 displays pseudocode for how the synthetic corpus was generated. 10,000 documents were generated with 50 tokens per document according to the generative story of a *dDMR* model where f was defined as a single-hidden-layer feedforward neural network with 5-dimensional hidden layer (sigmoid activation function) and a 100-dimensional output layer (softmax activation function). Each feature of the 100-dimensional document prior prefers 20%, 4 out of a total of 20 topics, on average, where each topic is sampled from a sparse Dirichlet prior over the vocabulary. The initialization of prior weights, η, in the *dDMR* model as well the the supervision for each document is outlined in Algorithm 4.

The observed document supervision was chosen such that it favored one feature in the 100-dimensional "true" feature space. We chose to generate document supervision this way, because this ensured that the "true" supervision for each document was sparse, preferring a small subset of topics, and was therefore more interpretable. The features are adjusted to this end by a technique similar to the neural network visualization technique in Simonyan, Vedaldi, and Zisserman (2013) or image perturbation in Deep Dream[3]. In this work, the input features are adjusted via gradient descent in order to better match the target output layer activations (encoding which topics are preferred by the current document). We generate corpora where the supervision noise variance ϵ varied from 0.01 to 10.0, and the observed supervision dimensionality d_{i} varied from 10 to 10,000 when generating corpora. This was to gauge how sensitive *DMR* was to noisy or high-dimensional supervision compared to *dDMR*.

[3]Demo: `https://deepdreamgenerator.com` ; code: `https://github.com/google/deepdream`

Algorithm 3 Generate synthetic corpus from *dDMR* model

Require: ϵ, d_i ◁ Variance on supervision noise, supervision dimensionality

$M \leftarrow 10^4$ ◁ No. of documents

$N_m \leftarrow 50$ ◁ No. of tokens/document

$K \leftarrow 20$ ◁ No. of topics

$V \leftarrow 100$ ◁ Vocabulary size

$d_o \leftarrow 100$ ◁ "True" supervision width

$d_h \leftarrow 5$ ◁ Hidden layer width

$\alpha_i, f \leftarrow GenDocSup(M, d_i, d_h, d_o)$ ◁ Initialize prior network and supervision (Alg 4)

$\alpha_o \leftarrow f(\alpha_i)$

$\alpha_i \leftarrow \alpha_i + Normal^{M \times d_i}(0.0, \epsilon)$

$\delta \sim Bernoulli^{d_o \times K}(0.2)$ ◁ Each true feature prefers roughly 20% of topics

for $k \leftarrow 1 \ldots K$ **do**

  $\phi_k \sim Dirichlet^V(0.1)$ ◁ Topic-word distribution drawn from sparse symmetric Dirichlet prior

end for

$\widetilde{\theta} \leftarrow \alpha_o^T \delta$ ◁ Document-topic prior

$D_{\text{index}}, W_{\text{index}} \leftarrow [], []$ ◁ Containers to store corpus

for $m \leftarrow 1 \ldots M$ **do**

  $\theta_m \sim Dirichlet(\widetilde{\theta}_m)$

  for $n \leftarrow 1 \ldots N$ **do**

    $D_{\text{index}} \leftarrow D_{\text{index}} + [m]$

    $z \sim Multinomial(\theta_m)$ ◁ Sample topic

    $w \sim Multinomial(\phi_z)$ ◁ Sample word

    $W_{\text{index}} \leftarrow W_{\text{index}} + [w]$

  end for

end for

return $(D_{\text{index}}, W_{\text{index}}, \alpha_i)$

Algorithm 4 Generate supervision and *dDMR* prior network.

Require: $M, d_i, d_h, d_o, \epsilon$ ◁ See Alg 3 for argument description.

$\alpha_i \sim Normal^{D \times d_i}(0.0, \epsilon)$ ◁ Initialize supervision from mean-zero Gaussian

for $m \leftarrow 1 \ldots M$ **do**

  $j \sim Uniform(\{1, 2, \ldots, d_o\})$ ◁ Sample "true" one-hot supervision

  $\alpha_o^m \leftarrow e_j$

end for

$W_i \sim Normal^{d_i \times d_h}(0.0, 1.0)$

$W_h \sim Normal^{d_h \times d_o}(0.0, 1.0)$

$f \leftarrow (\alpha) \mapsto \sigma(\sigma(\alpha W_i) W_h)$ ◁ Initialize prior network

$\mathcal{L} = -\sum_{m=1}^{M} \sum_{j=1}^{d_o} \alpha_o^{m,j} \log f(\alpha_i^m)_j$ ◁ categorical cross-entropy loss

for $i \leftarrow 1 \ldots 500$ **do**

  $\alpha_i \leftarrow \alpha_i - \frac{\delta \mathcal{L}}{\delta \alpha_i}$ ◁ Update input features to match output layer activation

end for

return (α_i, f)

Model Fit to Synthetic Corpora

For each corpus, we fit three 20-topic models: *LDA*, *DMR*, and *dDMR*. The *dDMR* model had an identical prior architecture to the generating model, but with randomly initialized weights. We fit models by the procedure described in Section 4.1.2 and evaluated model fit by heldout perplexity after 1,000 Gibbs sampling iterations[4].

Figure 4.4 displays the absolute difference in heldout perplexity between *DMR* and *dDMR* as a function of supervision dimensionality and noise. In the case when $d_i = 10,000$ and $\epsilon = 10.0$, *DMR* failed to train properly. NaNs were introduced in the gradient when performing hyperparameter optimization and the model could not be fit whereas *dDMR* did not exhibit this problem during training. *dDMR* always achieves at least as good model fit as *DMR* for the other synthetic datasets, with the gap between the two supervised models widening as the observed supervision both grows in dimensionality and becomes noisier.

[4]Gradient updates were performed after a burnin of 100 iterations. Prior hyperparameters were updated with adaptive learning rate algorithm Adadelta, with a base step size of $\eta = 0.5$ and $\rho = 0.95$.

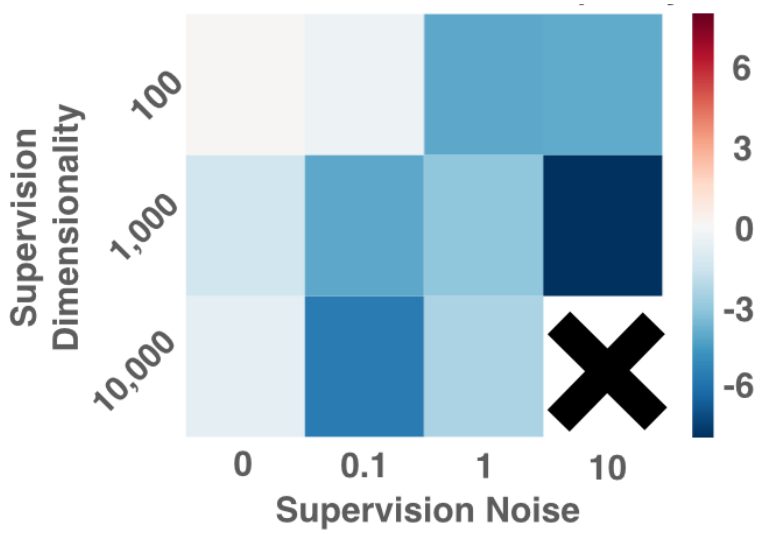


Figure 4.4: Difference between *dDMR* and *DMR* heldout perplexity for different synthetic corpora (varying supervision dimensionality and Gaussian noise). Bluer cells mean that *dDMR* achieved lower perplexity than *DMR*. The case where *DMR* hyperparameter optimization failed is marked by an "X".

Figure 4.5 displays training curves for each model with both train and heldout perplexity. For this corpus, *DMR* actually *underperforms LDA*, meaning that the noisy supervision was leading the document-topic priors astray. *dDMR*, on the other hand, can exploit the noisy supervision to achieve a much lower perplexity. *dDMR* achieves no worse heldout perplexity than *DMR* across all corpora, excelling when noise is high and supervision is wide. This suggests that *dDMR* is a promising model for using high-dimensional, noisy supervision such as user features to improve topic model fit.

4.3 *dDMR* Evaluation

We explore the flexibility of *dDMR* by considering three different datasets that include different types of metadata associated with each document. We first describe the

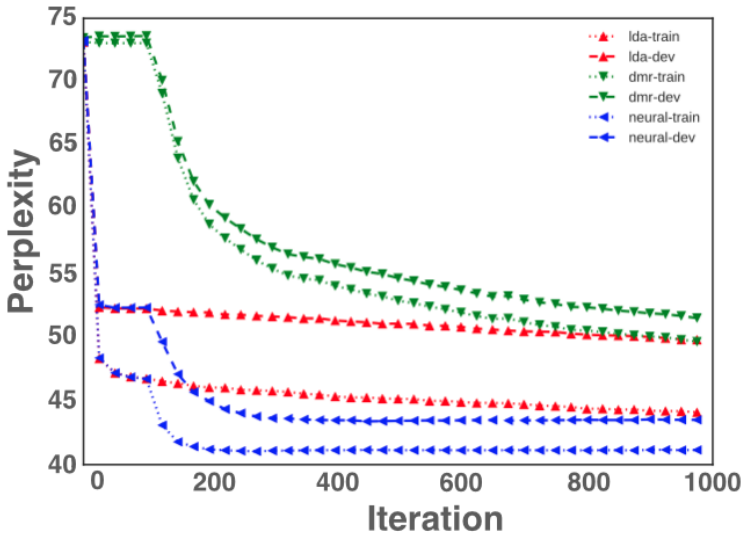


Figure 4.5: Training and heldout perplexity training curves for synthetic corpus generated with $d_i = 1000$ and $\epsilon = 1.0$ for each model. Training perplexity is marked by dotted lines, heldout by wider dashed lines. Models – *LDA*: green, *DMR*: red, *dDMR*: blue. The steep drop in perplexity after 100 iterations marks the end of burn-in and when hyperparameter optimization begins.

documents and metadata associated with each dataset and then the criteria by which we evaluate topic models.

4.3.1 Data

All datasets were preprocessed similarly. Article text was tokenized by non-alphanumeric characters and numerals were replaced by a special number token and infrequent word types were excluded from the corpora, although the number of word types kept varies slightly between corpora.

New York Times

The New York Times Annotated Corpus (Sandhaus, 2008) contains articles with extensive metadata used for indexing by the newspaper. For supervision, we used

the "descriptor" tags associated with each article assigned by archivists. These tags reflect the topic of an article, as well as organizations or people mentioned in the article. We selected all articles published in 1998, and kept those tags that were associated with at least 3 articles in that year – 2424 unique tags. 20 of the 200 most frequent tags were held out from training for validation purposes: { *"education and schools", "law and legislation", "advertising", "budgets and budgeting", "freedom and human rights", "telephones and telecommunications", "bombs and explosives", "sexual harassment", "reform and reorganization", "teachers and school employees", "tests and testing", "futures and options trading", "boxing", "firearms", "company reports", "embargoes and economic sanctions", "hospitals", "states (us)", "bridge (card game)",* and *"auctions"*}. Articles contained an average of 2.1 tags each, with 738 articles not containing any of these tags. Tags were represented using a one-hot encoding to use for supervision.

Words occurring in more than 40% of documents were removed, and only the 15,000 most frequent types were retained. This resulted in a total of 89,397 articles with an average length of 158 tokens per article.

Amazon Product Reviews

The Amazon product reviews corpus (McAuley and Yang, 2016) contains reviews of products as well as images of the product. We sampled 100,000 Amazon product reviews: 20,000 reviews sampled uniformly from the *Musical Instruments*, *Patio, Lawn, & Garden*, *Grocery & Gourmet Food*, *Automotive*, and *Pet Supplies* product categories. We hypothesize that knowing information about the product's appearance will indicate which words appear in the review, especially for product images occurring in these categories. 66 of the reviews we sampled contained only highly infrequent

tokens, and were therefore removed from our data, leaving 99,934 product reviews. Articles were preprocessed identically to the New York Times data.

We include images as supervision by passing each product's image through the Caffe convolutional neural network reference model, trained to predict ImageNet object categories[5]. We then extract the 4096-dimensional second fully-connected layer from this network to use as document supervision. Using these features as supervision in a *dDMR* model with a feedforward network prior is similar to fine-tuning a pretrained CNN to predict a new set of labels. Since the Caffe reference model is already trained on a large corpus of images, we chose to fine-tune only the final layers so as to learn a transformation of the already learned representation.

Reddit Messages

We finally constructed a corpus of online text by selecting a sample of Reddit posts made in January 2016. A standard stop list was used to remove frequent function words and we restricted the vocabulary to the 30,000 most frequent types. We restricted posts made to subreddits, collections of topically-related threads, with at least ten comments in this month (26,830 subreddits), and made by users with at least five comments across these subreddits (total of 1,351,283 million users). We then sampled 10,000 users uniformly at random and used all their comments as a corpus, for a total of 389,234 comments over 7,866 subreddits (document length mean: 16.3, median: 9) We considered a one-hot encoding of the subreddit ID a comment belonged to as supervision.

This corpus differs from the others in two ways. First, Reddit documents are very short, which presents a challenge for topic models that rely on detecting correlations

[5]Features used directly from `http://jmcauley.ucsd.edu/data/amazon/`

in token use within a document. Second, the Reddit metadata that may be useful for topic modeling is necessarily high-dimensional (e.g. subreddit identity, a proxy for topical content), so we believed that *DMR* will likely have trouble exploiting it.

4.3.2 Experiment Description

We used the same procedure to fit topic models on each dataset. Hyperparameter gradient updates were performed after a burnin period of 100 Gibbs sampling iterations. Hyperparameters were updated with the adaptive learning rate algorithm Adadelta with a tuned base learning rate and fixed $\rho = 0.95$[6]. All models were trained for a maximum of 15,000 epochs, with early stopping if heldout perplexity showed no improvements after 200 epochs (evaluated once every 20 epochs).

We used single-hidden-layer multi-layer perceptrons (MLPs), with rectified linear unit (ReLU) activations on the hidden layer, and linear activation on the output layer for the *dDMR* neural prior architecture. We sampled three architectures for each dataset, by drawing layer widths independently at random from $[10, 500]$, and also included two architectures with $(50, 10)$ and $(100, 50)$, *(hidden, output)* layers [7]. We compare the performance of *dDMR* to *DMR* trained on the same feature set as well as *LDA*.

For the New York Times dataset, we also compare *dDMR* to *DMR* trained on features after applying principal components analysis (PCA) to reduce the dimensionality of descriptor feature supervision, sweeping over PCA projection width in

[6] We found this adaptive learning rate algorithm improved model fit in many fewer iterations than gradient descent with tuned step size and decay rate for all models.

[7] We included these two very narrow architectures to ensure that some architecture learned a small feature representation, generalizing better when features are very noisy or only provide a weak signal for topic modeling. We restricted ourselves to only train *dDMR* models with single-hidden-layer MLPs in the priors to limit our search space.

$\{10, 50, 100, 250, 500, 1000\}$. Comparing performance of *dDMR* to PCA-reduced *DMR* tests two modeling choices. First, it tests the hypothesis that explicitly learning a representation for document annotations to maximize data likelihood produces a "better-fit" topic model than learning this annotation representation in unsupervised fashion – a two-step process. It also lets us determine if a linear dimensionality reduction technique is sufficient to learning a good feature representation for topic modeling, as opposed to learning a non-linear transformation of the document supervision. Note that we cannot apply PCA to reduce the dimensionality for subreddit id in the Reddit data, since these are one-hot features.

Model Selection Documents in each dataset were partitioned into ten equally-sized folds. Model training parameters of ℓ_1 and ℓ_2 regularization penalties on feature weights for *DMR* and *dDMR* and the base learning rate for each model class were tuned to minimize heldout perplexity on the first fold. These were tuned *independently for each model*, with number of topics fixed to 10, and *dDMR* architecture fixed to narrow layer widths $(50, 10)$. Model selection was based on the macro-averaged performance on the next eight folds, and we report performance on the remaining fold. We selected models separately for each evaluation metric. For *dDMR*, model selection amounts to selecting the document prior architecture, and for *DMR* with PCA-reduced feature supervision, model selection involved selecting the PCA projection width.

4.3.3 Evaluation

Each model was evaluated according to heldout (1) perplexity, (2) topic coherence by normalized pointwise mutual information (NPMI) (Lau, Newman, and Baldwin, 2014), and (3) a dataset-specific predictive task. We finally collect user preferences

for topics learned by each model. These are all typical approaches to evaluating topic models (Paul, 2015a).

NPMI computes an automatic measure of topic quality: the sum of pointwise mutual information between pairs of the m most likely words normalized by the negative log probability of each pair jointly occurring within a document (Equation 4.6). A topic with a large NPMI score is one whose most probable words tend to occur in the same documents more frequently than chance. We calculated this topic quality metric on the top 20 most probable words in each topic, and averaged over the most coherent 1, 5, 10, and all learned topics. However, models were selected to only maximize average NPMI over all topics.

$$NPMI = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \frac{\log \frac{P(w_i, w_j))}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)} \tag{4.6}$$

For the prediction tasks, we used the sampled topic distribution associated with a document, averaged over the last 100 iterations, as features to predict a document label. For New York Times articles we predicted 10 of the 200 most frequent descriptor tags restricting to articles with exactly one of these descriptors. For Amazon, we predicted the product category a document belonged to (one of five), and for Reddit we predicted a heldout set of document subreddit IDs. In the case of Reddit, these heldout subreddits were 10 out of the 100 most prevalent in our data, and were held out just as in the New York Times prediction task. SVM models were fit on inferred topic distribution features and were then evaluated according to accuracy, F1-score, and area under the ROC curve. The SVM slack parameter was tuned by 4-fold cross-validation on 60% of the documents, and evaluated on the remaining 40%.

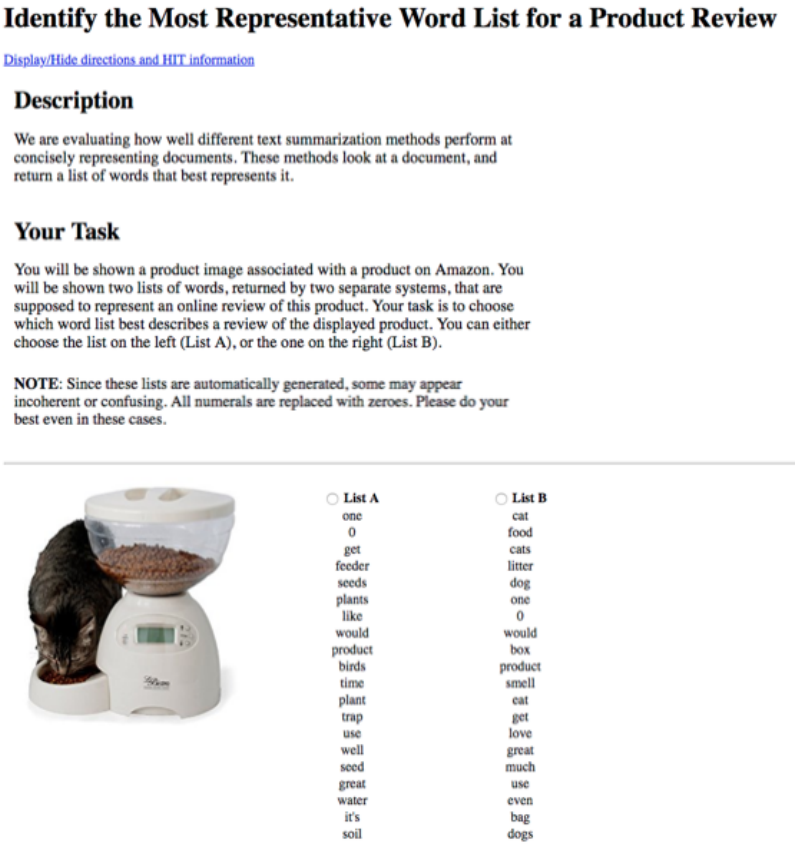


Figure 4.6: Screenshot of the topic quality judgment HIT. Here we elicit which of two topics humans believe is more likely for an Amazon product with the displayed image (a cat feeder).

We also collected human topic judgments using Amazon Mechanical Turk (Callison-Burch and Dredze, 2010). Each subject was presented with a human-readable version of the features used for supervision. For New York Times articles we showed the descriptor tags, for Amazon the product image, and for Reddit the name, title, and public description of the subreddit. We showed the top twenty words for the most probable topic sampled for the document with those features, as learned by two different models. One topic was learned by *dDMR* and the other was either learned by either *LDA* or *DMR*. The topics presented were from the 200-topic model architecture that maximized NPMI on development folds. Annotators were asked "*to choose which word list best describes a document ...*" with the displayed features. The topic learned by *dDMR* was shuffled to lie on either the right or left for each Human Intelligence Task (HIT). An example HIT for the Amazon data is shown in Figure 4.6.

We obtained judgments on 1,000 documents for each dataset and each model evaluation pair – 6,000 documents in all. This task can be difficult for many of the features, which may be unclear (e.g. descriptor tags without context) or difficult to interpret (e.g. images of unfamiliar automotive parts). We chose to not present the document text as well, since we did not want subjects to evaluate topic quality based on token overlap with the actual document.

4.3.3.1 Model Fit

dDMR achieves lower perplexity than *LDA* or *DMR* for most combinations of number of topics and dataset (Table 4.1). It is striking that *DMR* achieves higher perplexity than *LDA* in many of these conditions. This is particularly true for the Amazon dataset, where *DMR* consistently lags behind *LDA*. *Supervision alone does not improve topic*

Z	Model	NYT		Amazon		Reddit	
10	*LDA*	3429	(5)	2300	(7)	3811	(15)
	DMR	**3385**	(6)	2475	(9)	3753	(10)
	DMR-PCA	3417	(8)				
	dDMR	3395	(7)	**2272**	(68)	**3624**	(13)
20	*LDA*	3081	(6)	2275	(7)	3695	(19)
	DMR	3018	(4)	2556	(48)	3650	(8)
	DMR-PCA	3082	(8)				
	dDMR	**3023**	(7)	**2222**	(7)	**3581**	(16)
50	*LDA*	2766	(8)	2269	(9)	3695	(17)
	DMR	2797	(34)	2407	(20)	3640	(40)
	DMR-PCA	2773	(9)				
	dDMR	**2657**	(8)	**2197**	(13)	**3597**	(17)
100	*LDA*	2618	(8)	2246	(10)	3676	(19)
	DMR	2491	(27)	2410	(75)	3832	(30)
	DMR-PCA	2644	(52)				
	dDMR	**2433**	(10)	**2215**	(6)	**3642**	(18)
200	*LDA*	2513	(8)	2217	(7)	3653	(19)
	DMR	2630	(13)	2480	(65)	3909	(15)
	DMR-PCA	2525	(14)				
	dDMR	**2394**	(9)	**2214**	(12)	**3587**	(11)

Table 4.1: Test fold heldout perplexity for each dataset and model for number of topics Z. Standard error of mean heldout perplexity over all cross-validation folds in parentheses.

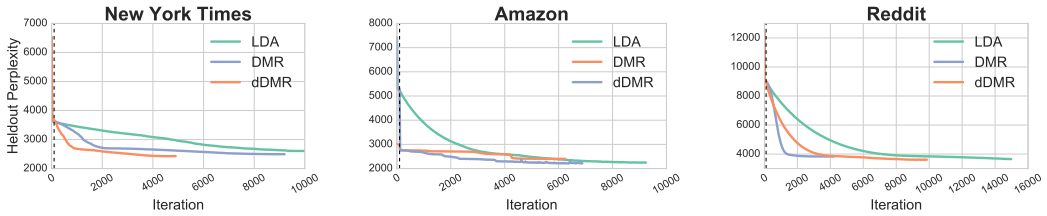


Figure 4.7: Heldout perplexity as a function of iteration for lowest-perplexity models with $Z = 100$. The vertical dashed line indicates the end of the burn-in period and when hyperparameter optimization begins.

model fit if it is too high-dimensional for learning. Perplexity is higher on the Reddit data for all models due to both a larger vocabulary size and shorter documents.

It is also worth noting that finding a low-dimensional linear projection of the supervision features with PCA does not improve model fit as well as *dDMR*. *dDMR* benefits both from joint learning to maximize corpus log-likelihood and possibly by the flexibility of learning non-linear projection (through the hidden layer ReLU activations).

Another striking result is the difference in speed of convergence between the supervised models and *LDA* (Figure 4.7). Even supervision that provides a weak signal for topic modeling, such as Amazon product image features, can speed convergence over *LDA*. In certain cases (Figure 4.7 left), training *dDMR* for 1,000 iterations results in a lower perplexity model than *LDA* trained for over 10,000 iterations.

In terms of actual run time, parallelization of model training differs between supervised models and *LDA*. Gradient updates necessary for learning the representation can be trivially distributed across multiple cores using optimized linear algebra libraries (e.g. BLAS), mitigating the additional cost incurred by hyperparameter updates in supervised models. In contrast, the Gibbs sampling iterations can also be parallelized,

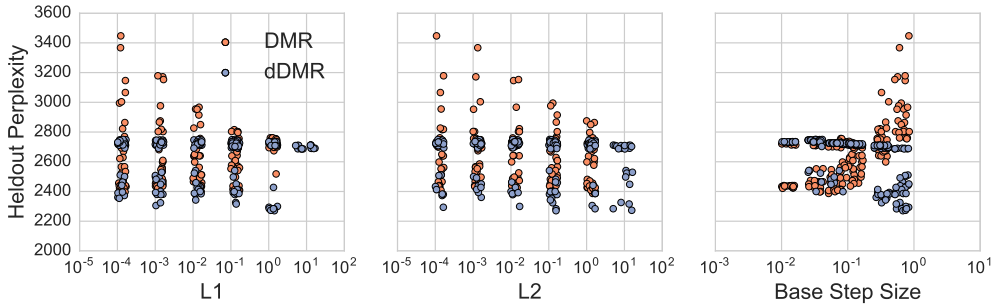


Figure 4.8: Heldout perplexity on the Amazon data tuning fold for *DMR* (orange) and *dDMR* (purple) with a (50, 10) layer architecture as a function of training parameters: ℓ_1, ℓ_2 feature weight regularization, and base learning rate. All models were trained for a fixed 5,000 iterations with horizontal jitter added to each point.

but not as easily, ultimately making resampling topics the most expensive step in model training. Because of this, the potential difference in runtime for a single iteration between *dDMR* and *LDA* is small, with the former converging in far fewer iterations. The time taken per iteration by *DMR* or *dDMR* was at most twice as long as *LDA* across all experiments.

Sensitivity to Learning Parameters Also, *dDMR* performance is much less sensitive to training parameters relative to *DMR*. While *DMR* requires heavy ℓ_1 and ℓ_2 regularization and a very small step size to achieve low heldout perplexity, *dDMR* is relatively insensitive to the penalty on regularization and benefits from a higher base learning rate (Figure 4.8). We found that dDMR *is easier to tune than* DMR, requiring less exploration of the training parameters. This is also corroborated by higher variance in perplexity achieved by *DMR* across different cross-validation folds (Table 4.1).

		New York Times				Amazon				Reddit			
Z	Model	1	5	10	Overall	...				...			
10	*LDA*	52	49	43	43	25	23	20	20	**125**	**82**	**56**	**56**
	DMR	53	50	42	42	**58**	**43**	**31**	**31**	43	35	30	30
	DMR-PCA	**63**	**53**	**45**	**45**								
	dDMR	57	51	44	44	24	21	19	19	109	62	46	46
20	*LDA*	62	59	54	45	27	25	23	20	**121**	**87**	**59**	**42**
	DMR	63	60	56	45	66	56	**53**	**43**	81	49	41	34
	DMR-PCA	**76**	**61**	**57**	**47**								
	dDMR	69	60	55	45	**97**	**61**	**53**	40	109	66	49	38
50	*LDA*	80	66	62	44	30	27	25	20	**135**	**96**	**64**	34
	DMR	80	**67**	**63**	**46**	**136**	**81**	**73**	**58**	51	46	41	33
	DMR-PCA	**82**	**67**	63	45								
	dDMR	76	65	61	45	71	65	62	44	121	74	54	**36**
100	*LDA*	77	71	66	40	58	34	30	20	135	74	54	31
	DMR	**80**	**74**	70	**45**	**147**	**83**	**75**	**59**	111	67	50	**34**
	DMR-PCA	79	69	**75**	**45**								
	dDMR	77	73	68	44	68	67	66	55	**135**	**78**	**55**	31
200	*LDA*	78	74	70	36	60	39	34	18	**135**	**100**	**67**	29
	DMR	91	**76**	80	42	69	67	67	**61**	132	84	59	**32**
	DMR-PCA	**94**	**76**	**81**	42								
	dDMR	78	70	66	**45**	**85**	**73**	**69**	39	**135**	87	61	30

Table 4.2: Top-1, 5, 10, and overall topic NPMI across all datasets. Models that maximized overall NPMI across dev folds were chosen and the best-performing model is in bold.

4.3.3.2 Topic Quality

Results for the automatic topic quality evaluation, NPMI, are mixed across datasets. In many cases, *LDA* and *DMR* score highly according to NPMI, despite achieving higher heldout perplexity than *dDMR* (Table 4.2). This may not be surprising as previous work has found that perplexity does not correlate well with human judgments of topic coherence (Lau, Newman, and Baldwin, 2014).

However, in the Mechanical Turk evaluation, subjects found that *dDMR*-learned topics are more representative of document annotations than *DMR* (Table 4.3). While

	LDA	*DMR*
New York Times	51.1%	51.9%
Amazon	51.9%	61.4%*
Reddit	55.5%*	57.6%*

Table 4.3: % HITs where humans considered *dDMR* topics to be more representative of document supervision than the competing model. $*$ denotes statistical significance according to a one-tailed binomial test at the $p = 0.05$ level.

subjects only statistically significantly favored *dDMR* models over *LDA* on the Reddit data, they favored *dDMR* topics over *LDA* by a small margin across all datasets, and statistically significantyl preferred *dDMR* topics over *DMR* on two of the three datasets. This is contrary to the model rankings according to NPMI, which predict that *DMR* topics would be preferable.

4.3.3.3 Predictive Performance

Finally, we consider the utility of the learned topic distributions for downstream prediction tasks, a common use of topic models. Although token perplexity is a standard measure of topic model fit, it has no direct relationship with how topic models are typically used: to identify consistent themes or reduce the dimensionality of a document corpus. We found that features based on topic distributions from *dDMR* outperform *LDA* and *DMR* on the Amazon and Reddit data when the number of topics fit is large, although they fail to outperform *DMR* on New York Times (Table 4.4). Heldout perplexity is strongly correlated with predictive performance, with a Pearson correlation coefficient, $\rho = 0.898$ between F1-score and heldout perplexity on the Amazon data. This strong correlation is likely due to the tight relationship between words used in product reviews and product category: a model that assigns high likelihood to a words in a product review corpus should also be informative of the

		New York Times			Amazon			Reddit		
Z	Model	F1	Accuracy	AUC	...			...		
10	*LDA*	0.208	**0.380**	0.767	**0.662**	**0.667**	**0.891**	0.130	0.276	0.565
	DMR	0.236	0.367	0.781	0.311	0.407	0.619	0.092	0.229	**0.597**
	DMR-PCA	**0.280**	0.347	0.758						
	dDMR	0.154	0.347	**0.790**	0.608	0.656	0.864	**0.170**	**0.300**	0.596
20	*LDA*	0.315	0.463	0.784	0.657	0.659	0.887	**0.121**	0.258	**0.579**
	DMR	0.319	0.477	0.805	0.294	0.405	0.647	0.057	0.245	0.520
	DMR-PCA	0.343	**0.540**	**0.831**						
	dDMR	**0.424**	0.523	0.797	**0.706**	**0.711**	**0.911**	0.071	**0.274**	0.566
50	*LDA*	0.455	0.613	0.849	0.630	0.634	0.870	0.131	0.199	0.542
	DMR	0.478	0.650	0.877	0.396	0.499	0.619	**0.145**	0.261	**0.580**
	DMR-PCA	0.505	**0.667**	**0.887**						
	dDMR	**0.507**	0.657	0.856	**0.716**	**0.726**	**0.916**	0.118	**0.272**	0.551
100	*LDA*	0.531	0.657	0.874	0.646	0.649	0.874	0.148	0.201	0.538
	DMR	0.552	0.683	0.898	0.392	0.463	0.688	0.107	0.233	0.512
	DMR-PCA	**0.602**	**0.687**	**0.917**						
	dDMR	0.514	0.653	0.893	**0.650**	**0.660**	**0.893**	**0.172**	**0.316**	**0.614**
200	*LDA*	0.566	0.683	0.903	0.646	0.651	0.882	0.111	0.227	0.517
	DMR	0.576	0.670	**0.917**	0.288	0.401	0.697	0.089	0.229	0.499
	DMR-PCA	**0.648**	**0.762**	0.915						
	dDMR	0.605	0.730	0.903	**0.716**	**0.721**	**0.909**	**0.198**	**0.323**	**0.580**

Table 4.4: Top F-score, accuracy, and AUC on prediction tasks for all *dDMR* evaluation datasets.

product categories. Prior work showed that upstream supervised topic models, such as *DMR*, learn topic distributions that are effective at downstream prediction tasks (Benton et al., 2016b). We find that topic distributions learned by *dDMR* improve over *DMR* in certain cases, particularly as the number of topics increases.

4.3.3.4 Qualitative Results

We also qualitatively explored the product image representations *DMR* and *dDMR* learned on the Amazon data. To do so, we computed and normalized the prior document distribution for a sample of documents learned by the lowest perplexity *DMR* and *dDMR* 200-topic models:

$$p(k|m) = \frac{\widetilde{\theta}_m}{\sum_{k=1}^{Z} \widetilde{\theta}_{m,k}} \tag{4.7}$$

This is the prior probability of sampling topic k conditioned on the features for document m (before seeing any words in the document). We then marginalize over topics to yield the conditional probability of a word w given document m: $p(w|m) = \sum_{k=1}^{Z} p(w|k)p(k|m)$.

Table 4.5 contains a sample of these probable words given document supervision. We find that *dDMR* identifies words likely to appear in a review of the product pictured. However, some images lead *dDMR* down a garden path. For example, a bottle of "Turtle Food" should not be associated with words for human consumables like "coffee" and "chocolate", despite the container resembling some of these products. However, the image-specific document priors *DMR* learned are not as sensitive to the actual product image as those learned by *dDMR*. The prior conditional probabilities $p(w|m)$

for “Turtle Food”, “Slushy Magic Cup”, and “Rawhide Dog Bones” product images are all ranked identically by *DMR*.

4.4 Application: Predicting Policy Surveys with Twitter Data

In section 4.2, we presented a new supervised topic model that is more resilient to noisy supervision than *DMR*. In this section we apply *DMR* and *dDMR* to three different Twitter public policy opinion datasets, comparing how models conditioned on inferred user location features fare against supervised models trained with distant demographics and policy-relevant features.

4.4.1 Motivation

One goal of social media analytics is to complement or replace traditional survey mechanisms (Thacker and Berkelman, 1988; Krosnick, Judd, and Wittenbrink, 2005). Traditional phone surveys are both slow and expensive to run. For example, the CDC’s annual Behavioral Risk Factor Surveillance System (BRFSS) is a health-related telephone survey that collects health data by calling more than 400,000 Americans. The survey costs millions of dollars to run each year, so adding new questions or obtaining finer-grained temporal information can be prohibitive.

We consider three different public opinion data Twitter datasets. Each of these datasets consists of tweets that are relevant to a BRFSS survey question, an annual phone survey of hundreds of thousands of American adults, chosen for its very large and geographically widespread sample. We selected the following three BRFSS questions: the percentage of respondents in each U.S. state who (1) have a firearm

Image	Item	*dDMR* Probable Words	*DMR* Probable Words
	**Guitar Foot Rest**	**grill** easy cover well fit **mower** fits **job gas hose** light **heavy easily stand back** nice works **use enough pressure**	fit easy well works **car** light **sound quality work guitar would 0000** cover nice **looks bought install battery 00** fits
	**Bark Collar**	fit battery 0000 light install car sound easy work **unit amp** 00 **lights mic power** works **000 took replace installed**	fit easy **well** works car light work **quality** sound **would guitar** 0000 **cover nice bought looks** install battery 00 **fits**
	**Turtle Food**	taste coffee flavor food like love cat tea product tried dog eat chocolate **litter** cats **good best bag** sugar loves	taste coffee dog like love flavor food cat product tea cats tried **water dogs** loves eat chocolate **toy mix** sugar
	**Slushy Magic Cup**	food taste cat coffee flavor love like dog tea **litter** cats eat tried product chocolate loves **bag** good **best smell**	taste coffee dog like love flavor food cat product tea cats tried **water dogs** loves eat chocolate **toy mix** good
	**Rawhide Dog Bones**	food cat dog cats **litter** dogs loves love product **smell** eat **box** tried **pet bag hair** taste **vet** like **seeds**	taste **coffee** dog like love **flavor** food cat product **tea** cats tried **water** dogs loves eat **chocolate toy mix good**
	**Instrument Cable**	sound **amp** guitar **mic pedal sounds price volume** quality **cable great bass microphone strings** music **play recording 000 tone** unit	sound guitar **fit easy well 0000 works car** quality **light** music **cover work one set nice looks 00 install** unit

Table 4.5: Top twenty words associated with each of the product images – learned by *dDMR* vs. *DMR* ($Z = 200$). These images were drawn at random from the Amazon corpus (no cherry-picking involved). Word lists were generated by marginalizing over the prior topic distribution associated with that image and then normalizing each word's probability by subtracting off its mean marginal probability *across all images in the corpus*. This is done to avoid displaying highly frequent words. Words that differ between each model's ranked list are in bold.

Guns	Vaccines	Smoking
gun, guns, second amendment, 2nd amendment, firearm, firearms	vaccine, vaccines, vaccinate, vaccinated	smoking, smoked, tobacco, cigarettes, cigarette, cigar, smoker, smokers

Table 4.6: The keyphrases used to filter the BRFSS-related Twitter policy datasets.

in their house (data from 2001, when the question was last asked), (2) have had a flu shot in the past year (from 2013), and (3) are current smokers (from 2013).

We would like to fit topic models to these data, and use the inferred topic distribution to predict survey responses at the state level. We consider two classes of supervision for guiding supervised topic models: weak author demographic and opinion supervision based on the inferred location of the tweet author. We also compare how predictive of BRFSS survey responses *DMR* is to *dDMR* when we use a one-hot encoding of the author's inferred location as topic model supervision – either at the state, county, or the city level.

4.4.2 Datasets

We created three Twitter datasets based on keyphrase filtering (Table 4.6) with data collected from Dec. 2012 through Jan. 2015 to match tweets relevant to these three survey questions. We selected 100,000 tweets uniformly at random for each dataset and geolocated them to state/county using *Carmen* (Dredze et al., 2013). Geolocation coverage is shown in Table 4.7.

We consider the following sources of (distant) topic model supervision along with one-hot author location indicators:

4.4.2.1 Survey

This indirect supervision uses the values of the BRFSS survey responses that we are trying to predict. Tweets whose authors are resolved to a state are assigned the proportion of "yes" survey respondents within that state. This setting reflects predicting the values for some states using data already available from other states. This setting is especially relevant for BRFSS, since the survey is run by each state with results collected and aggregated nationally. Since not all states run their surveys at the same time, BRFSS routinely has results available for some states but not yet others.

4.4.2.2 Census

We also experimented with an alternative indirect type of supervision: demographic information from the 2010 U.S. Census[8]. Demographic variables are correlated with the responses to the surveys we are trying to predict (Hepburn et al., 2007; King, Dube, and Tynan, 2012; Gust et al., 2008), so we hypothesize that conditioning on demographic information may lead to more predictive and interpretable topic models than no supervision at all. This approach may be advantageous when domain-specific survey information is not readily available.

From the Census, we used the percentage of white residents per county as supervision for tweets whose county could be resolved. Although this feature is not directly related to the survey proportions we are trying to predict, it is sampled at a finer granularity than the state-level survey feature. Proportion of tweets tagged with this feature are also included in Table 4.7. In our experiments we consider these

[8] `http://www.census.gov/2010census/data/`

Dataset	Vocab	State	County	City	BRFSS
Guns	12,358	29.7%	18.6%	16.7%	Owns firearm
Vaccines	13,451	23.6%	16.2%	14.8%	Had flu shot
Smoking	13,394	19.6%	12.8%	12.7%	Current smoker

Table 4.7: A summary of the three Twitter public policy datasets: size of the vocabulary, proportion of messages tagged at the state and county level, and the state-level survey question (BRFSS) asked.

two types of supervision in isolation to assess the usefulness of each class of distant supervision.

4.4.2.3 User Location Features

In addition, we consider conditioned models on a one-hot encoding of location. We consider three different levels of granularity: *state*, *county*, and *city*. We restricted to only locations that were resolved in the United States, treating tweets resolved to other countries as though they were not resolved at all. As the surveys we are trying to predict are specific to American opinions, this ensured that document-level features were restricted to those tweets that were more likely to come from United State residents. It also means that tweets that are tagged with a specific location are a strict subset of those that were tagged by the state-level Survey feature. Tweets that *Carmen* was unable to resolve were assigned a `NOT_RESOLVED` location feature, and finer granularity features backed off to the most specific type of location resolved.

We consider these direct user location features since like the Census feature it is agnostic to which survey question we are trying to predict. However, unlike the Census feature, a topic model conditioned directly on location has more flexibility to learn which topics are more likely in that specific location, rather than relying on a single, the proportion of white residents in the county, as a proxy.

4.4.3 Experiments

We fit *DMR* and *dDMR* models conditioned on each feature set, tuning for held-out perplexity and evaluated its ability to predict the survey proportion for each state. We also compared to an *LDA* model without any supervision.

The text was preprocessed by removing stop words and low-frequency words. We also removed usernames, URLs, and non-alphanumeric tokens. We applied z-score normalization to the BRFSS/Census values within each dataset, so that the mean value was 0. For tweets whose location could not be resolved, the Survey and Census document supervision was set to 0.0, and the `NOT_RESOLVED` one-hot location features was active.

Evaluation We evaluated the utility of topics as features for predicting the survey value for each U.S. state, reflecting how well topics capture themes relevant to the survey question. We inferred θ_m for each tweet and then averaged these topic vectors over all tweets originating from each state, to construct 50 feature vectors per model. We used these features in a regularized linear regression model. Average root mean-squared error (RMSE) was computed using five-fold cross-validation: 80% of the 50 U.S. states were used to train, 10% to tune the ℓ_2 regularization coefficient on the ridge regression model, and 10% were used for evaluation. In each fold, the topic models used supervision only for tweets from the training set states, while the $\boldsymbol{\alpha}$ values were set to 0.0 (a neutral value) for the held-out states.

For both perplexity and prediction performance, we sweep over number of topics in $\{10, 25, 50, 100\}$ and report the best result. Results are averaged across five sampling runs to mitigate variation in performance due to estimating model parameters by Gibbs sampling.

Model Selection For tuning, we held out 10,000 tweets from the guns dataset and used the best learning parameters for all datasets. We ran Spearmint (Snoek, Larochelle, and Adams, 2012) for 100 iterations to tune the learning parameters, running each sampler for 500 iterations. We used Spearmint since it allowed us to automatically explore a large space of learning parameters quickly without resorting to brute-force grid search. Spearmint was used to tune the following learning parameters: the initial values for $\boldsymbol{\omega}_b$ and $\boldsymbol{\eta}_b$, as well as ℓ_2 regularization on $\boldsymbol{\eta}_b$, $\boldsymbol{\omega}_b$, and $\boldsymbol{\eta}$.

Held-out perplexity is very sensitive to some parameters, such as initialization of $\boldsymbol{\eta}_b$ and $\boldsymbol{\omega}_b$, while other parameters, such as the ℓ_2 regularization on $\boldsymbol{\omega}_b$ had little effect. Once tuned, all models were trained for 2,000 iterations, using AdaGrad with a master step size of 0.02, with no hyperparameter updates made in the first 200 iterations.

4.4.3.1 Replication: Comparing *DMR* to *dDMR*

One crucial detail is that the initial set of experiments with *DMR* conditioned on Survey and Census features were run using a Java package, `sprite`[9], that implemented Sprite topic models – a class of upstream topic models with structured priors (Paul and Dredze, 2015). We attempted to replicate these experiments with a Python 3.5 library that supports defining and training *dDMR* models with feedforward neural network priors, `deep-dmr`[10]. Relying on `deep-dmr` was necessary as `sprite` does not support training *dDMR* models.

When replicating models in `deep-dmr`, we considered a different model selection scheme due to the wide space of possible *dDMR* models and time restrictions. For each model class conditioned on feature type, we performed a grid search on the held-out gun control tweets for ℓ_1 and ℓ_2 regularization constants in $\{0.0, 10^{-4}, 10^{-2}, 10^{-1}\}$

[9] `https://bitbucket.org/adrianbenton/sprite/`
[10] `https://github.com/abenton/deep-dmr`

and $\{10^{-4}, 10^{-2}, 10^{-1}, 10^{0}\}$, respectively. These constants were then applied to models trained on all datasets. We also swept over base learning rate for each model class in $\{10^{-2}, 10^{-1}, 10^{0}\}$, for Adadelta hyperparameter updates (this is the default hyperparameter update algorithm in this package). For all models, bias hyperparameters were initialized to $\boldsymbol{\eta_b} = -2$ and $\boldsymbol{\omega_b} = -4$, corresponding to sparse initial Dirichlet priors.

For *dDMR* and each feature set, we swept over three single-hidden-layer architectures with only linear activations for the document-topic prior: $\{[10, 5], [50, 10], [100, 50]\}$. This amounts to a simple lookup embedding of the state, county, or city features. For *DMR*, we use each of the feature sets as supervision, but for *dDMR* we only consider state, county, or city indicator features[11].

4.4.4 Results

We first present the results on comparing *DMR* conditioned on Survey and Census features to an unsupervised topic model, *LDA*. These experiments were run using the `sprite` package. We then present these experiments replicated using `deep-dmr`, with the new model learning and selection criteria as described in 4.4.3.1. We compare perplexity and predictive performance of conditioning on location features under this replication framework.

4.4.4.1 Evaluating Survey and Census Features

Results from training models in `sprite` are shown in Table 4.8. The important takeaway is that *DMR* conditioned on indirect user features are more predictive than LDA, an unsupervised model. Not only do the supervised models substantially reduce

[11] Conditioning a *dDMR* model with linear activations on a single feature offers no flexibility beyond *DMR* on that feature.

Features	Model	Guns		Vaccines		Smoking	
None	*LDA*	17.44	2313 ($\pm$52)	8.67	2524 ($\pm$20)	4.50	2118 ($\pm$5)
Survey	*DMR*	15.37	**1529** ($\pm$12)	6.54	**1552** ($\pm$11)	**3.41**	**1375** ($\pm$6)
Census	*DMR*	**11.51**	1555 ($\pm$27)	**5.15**	1575 ($\pm$90)	3.42	1377 ($\pm$8)

Table 4.8: RMSE of the prediction task (left) and average perplexity (right) of topic models over each dataset, $\pm$ the standard deviation (learned under `sprite`). Perplexity is averaged over 5 sampling runs and RMSE is averaged over 5 folds of U.S. states. As a benchmark, the RMSE on the prediction task using a bag-of-words model was 11.50, 6.33, and 3.53 on the Guns, Vaccines, and Smoking data, respectively.

prediction error, as might be expected, but they also have substantially lower perplexity, and thus seem to be topics that better represent the data.

The poor performance of *LDA* may be partially explained by the fact that Spearmint seems to overfit *LDA* to the tuning set. Other models attained a tuning set perplexity of between 1500 to 1600, whereas *LDA* attained 1200. To investigate this issue further, we separately ran experiments with hand-tuned models, which gave us better held-out results for *LDA*, though still worse than the supervised topic models (e.g., RMSE of 16.44 on the guns data). Although Spearmint tuning is not perfect, it is fair to all models.

For additional comparison, we experimented with a standard bag-of-words model, where features were normalized counts across tweets from each state. This comparison is done to contextualize the magnitude of differences between models, even though our primary goal is to compare different types of topic models. We found that the bag-of-words results (provided in the caption of Table 4.8) are competitive with the best topic model results. However, topic models are often used for other advantages, e.g., interpretable models.

Guns		Vaccines		Smoking	
$r = -1.04$	$r = 0.43$	$r = -0.25$	$r = 1.07$	$r = -0.62$	$r = 1.04$
gun	guns	ebola	truth	smoking	#cigar
mass	people	trial	autism	quit	#nowsmoking
shootings	human	vaccines	outbreak	stop	#cigars
call	get	promising	science	smokers	cigar
laws	would	experimental	know	#quitsmoking	james
democrats	take	early	connection	best	new
years	one	first	via	new	thank
since	away	results	knows	help	beautiful
australia	safe	hint	#mhealth	#smoking	#habanos
1996	use	safety	#tetanus	please	#cigarlovers

Table 4.9: Sample topics for the *DMR* model supervised with the survey feature. A topic with a strongly negative as well as a strongly positive η value was chosen for each dataset. Positive value indicates that the tweet originates from a state with many "yes" respondents to the survey question.

DMR conditioned on Census features yielded worse predictive performance than Survey-conditioned models on two of the three datasets, strangely enough. We found this surprising, but may be due to having an exceptionally small test set (test performance averaged over 5 sets of 10 examples/states each).

Qualitative Inspection Table 4.9 displays example topics learned by *DMR* conditioned on Survey features. For example, a topic about the results of the ebola vaccine trials is negatively correlated with vaccine refusal, while a topic about the connection between vaccines and autism is positively correlated with vaccine refusal. We did not observe noticeable qualitative differences in topics learned by the different models, with an exception of *LDA*, where the topics tended to contain more general words and fewer hashtags than topics learned by the supervised models.

Use Case: Predicting Support for Gun Restrictions We ran an additional experiment to consider the setting of predicting a new survey with limited available data. We

Features	Model	RMSE (2001 Y included)	RMSE (2001 Y omitted)
None	No model	7.26	7.59
	Bag of words	5.16	7.31
	LDA	6.40	7.59
Survey	*DMR*	**5.11**	**5.48**

Table 4.10: RMSE when predicting proportion respondents opposing universal background checks with topic distribution features. We experimented with (left) and without (right) including the 2001 proportion households with a firearm survey data as an additional feature. "*No model*" is the regression where we predict using only the 2001 proportion of households with a firearm.

chose the subject of requiring universal background checks for firearm purchases, a topic of intense interest in the U.S. in 2013 due to political events. Despite the national interest in this topic, telephone surveys were only conducted for less than half of U.S. states. We identified 22 individual state polls in 2013 that determined the proportion of respondents that opposed universal background checks. 15 of the states were polled by Public Policy Polling, while the remaining 7 states were polled by Bellwether Research, Nelson A. Rockefeller Research, DHM Research, Nielsen Brothers, Repass & Partners, or Quinnipiac University. We take this as a real-world example of our intended setting: a topic of interest where resources limited the availability of surveys.

We used a topic model trained with data from the universal background check (UBC) survey question as features for predicting the state values for the UBC surveys. As in the previous experiments, we used topic features in a linear regression model, sweeping over ℓ_2 regularization constants and number of topics, and we report test performance of the best-performing settings on the tuning set. We evaluated the model using five-fold cross-validation on the 22 states.

Additionally, we sought to utilize data from a previous, topically-related survey: the "Guns" BRFSS survey used in the previous section, which measured the proportion

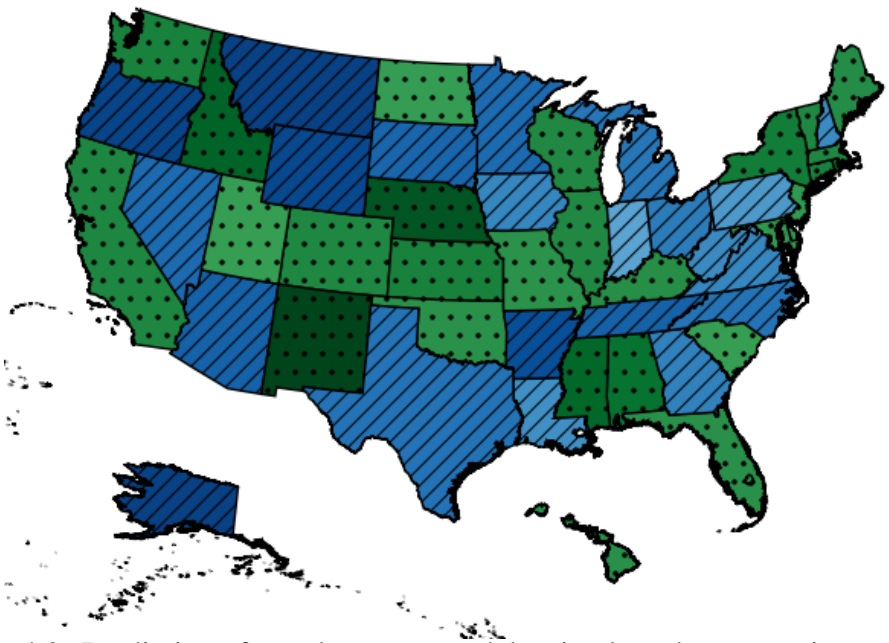


Figure 4.9: Predictions from the *DMR* model trained on the proportion opposed to universal background checks. The 22 blue states hatched with lines were in the model's training set, while we have no survey data for the 28 green, dotted states. Darker colors denote higher opposition to background checks. New Mexico is predicted to have the highest percent of respondents opposed (53%), while Utah has the lowest predicted opposed (18%).

of households with a firearm, asked in 2001. While the survey asks a different question, and is several years out of date, our hypothesis is that the results from the 2001 survey will be correlated with the new survey, and thus will be a good predictor. We experimented with and without including the values of the 2001 BRFSS survey (which is available for all 50 states) as an additional feature in the regression model.

Table 4.10 contains the cross-validation test results. We compared the supervised topic model performance to LDA as well as a bag-of-words model. To put the results in context, we also trained regression models using only the 2001 BRFSS values as features ("No model, 2001 Y included") as well as a regression model with no features at all, only an intercept ("No model, 2001 Y omitted").

In general, models that use text features outperform the baseline using only data from the 2001 survey, showing that text information derived from social media can improve survey estimation, even when using topically-related historic data. Moreover, the supervised *DMR* model trained on the UBC survey data is significantly better than an unsupervised topic model (*LDA*) with $p = 0.06$, under a paired t-test across folds. The difference between *DMR* and the bag-of-words model is not significant ($p = 0.16$), although the difference is larger in the setting where the 2001 survey data is omitted. For the Public Policy Polling surveys used to build the UBC data, the margin of error ranged from 2.9% (more than 1000 polled) to 4.4% (500 polled). An RMSE of 5.1 is approximately equivalent to a 10% margin of error at the 95% confidence level, equivalent to polling roughly 100 people.

Finally, we trained the *DMR* regression model (with 2001 BRFSS features) on all 22 states, and used this model to make predictions of opposition to universal background checks for the remaining 28 states. The predictions are shown in Figure 4.9. We generated similar plots for *dDMR* models conditioned on state and county features.

4.4.4.2 Conditioning on Location Features

Table 4.11 contains the performance of replicating the above perplexity and prediction evaluations with the model training and selection criteria described in Section 4.4.3.1. It also includes performance of *DMR* and *dDMR* models conditioned on one-hot location features.

There most salient finding is that all models perform within a single standard deviation of each other for all datasets and evaluation metrics. This is different than what we had observed originally where *LDA* was soundly beat by *DMR*. As mentioned

Features	Model	Guns		Vaccines		Smoking	
None	*LDA*	9.72	1341	6.89	2192	3.81	1608
Survey	*DMR*	9.76	1356	7.71	2216	5.24	1621
Census	*DMR*	12.11	1369	6.90	2223	3.64	1611
State	*DMR*	**8.26**	1370	6.89	2220	3.91	1608
State	*dDMR*	10.30	1350	8.00	2185	3.60	1624
County	*DMR*	11.75	1380	6.59	**2183**	3.75	1616
County	*dDMR*	11.75	1366	6.17	2193	3.75	1613
City	*DMR*	10.35	1366	7.40	2185	3.55	1608
City	*dDMR*	10.81	**1340**	**5.75**	2194	**3.47**	**1605**

Table 4.11: RMSE of the prediction task (left) and average perplexity (right) of topic models over each dataset as replicated in `deep-dmr`. *State*, *County*, and *City* are models trained with a one-hot encoding of the author's inferred state, county, or city.

above, *LDA* had been likely performing worse since Spearmint overfit to the tuning set.

Why is the Performance so Different? We took great pains to ensure that both `sprite` and `deep-dmr` optimized models identically. We made sure that initializing *LDA* under both both frameworks with the same topic samples yielded identical training and heldout perplexity, and that they achieved similar final heldout perplexity when learning over synthetic data. In the process of uncovering the difference between the original experiments and replications, we noticed two small discrepancies between these implementations that were subsequently resolved:

- Treating every other token *in the corpus* as heldout as opposed to every other token *within each document*. Since words are shuffled within each document as a preprocessing step, this did not affect heldout perplexity significantly.

- The bias hyperparameters were initialized to different values in each implementation: `deep-dmr` initialized them to $\boldsymbol{\eta_b} = -1$ and $\boldsymbol{\omega_b} = -2$.

The critical differences between model training and selection in the `sprite`-trained models and the `deep-dmr` replicated models are as follows:

- Hyperparameters updated with Adagrad (master learning rate fixed to 0.02) $\rightarrow$ Adadelta (tuned master learning rate).
- Spearmint-tuned model selection $\rightarrow$ Grid search for training parameters for each model class
- Swept for bias hyperparameter initialization $\rightarrow$ Fixed to $\boldsymbol{\eta_b} = -2$ and $\boldsymbol{\omega_b} = -4$ for all models.

Although these **should be** relatively minor choices, they clearly had a profound impact on the quality of topic models that were learned.

4.5 Summary

This chapter presents a non-traditional application of user features and embeddings: conditioning the topic distribution in a supervised topic model on user features. We show that supervision at the author-level is important for modeling short-text corpora such as collections of social media messages. Specifically, we show that modeling three different opinionated Twitter datasets benefit from distant, carefully chosen user features – responses to state-wide polls and county-level demographic features from the United States census.

Synthetic experiments show that *dDMR* is most appropriate when one is given very high-dimensional and noisy supervision. Empirically we find that it achieves significantly better model fit (according to heldout perplexity) than *DMR* on three datasets with high-dimensional supervision regardless of number of topics learned.

Topic distributions inferred by *dDMR* with only location indicator features perform on par with models conditioned on carefully selected survey and census features in three Twitter opinion datasets related to guns, vaccines, and smoking. However, the performance of models heavily depends on choices in model training and selection.

In light of our experiments, we encourage topic model practitioners to consider fitting supervised topic models with document-level user features instead of *LDA* when exploring new corpora. Even distant or high-dimensional supervision helps improve the model fit and topic quality, especially when choosing a *dDMR* model. Regardless, Gibbs samplers converge far faster for models superivsed by user features than unsupervised – a very practical reason to opt for fitting *dDMR* models over *LDA*.

Chapter 5

Multitask User Features for Mental Condition Prediction

In Chapter 3 we showed how multiview user embeddings can be learned from different views of user behavior and can then be used to predict hashtag use or friending behavior. In Chapter 4 we showed that user-level features can even be used to speed topic model convergence and improve topic model fit. Although making accurate predictions of who will friend whom is valuable to social media platform engineers and fitting topic models more quickly is valuable to social scientists understanding large text datasets, they are not strong examples of how user features can directly improve people's lives. In this chapter, we show how to train stronger neural classifiers to predict a Twitter user's risk for suicide as well as other mental health conditions solely from their tweets. We do this by fitting classifiers in the multitask learning (MTL) framework where we consider predicting multiple mental conditions a user has along with their gender as auxiliary tasks.

Section 5.1 describes the motivation for building mental condition classifiers from user tweets: why would we want to predict someone's mental condition from social media and how might this save lives? Section 5.2 describes the types of neural

classifier architectures we consider: basic logistic regression, single-task feedforward, and MTL feedforward architectures. Section 5.3 describes the training and evaluation dataset, a collection of users with self-reported mental condition along with age and gender-matched control users. Section 5.4 describes the experiment protocol and how model hyperparameters were chosen, a crucial step in any careful comparison of model classes. Section 5.5 ends by presenting model performance at mental condition identification and an ablation analysis of which user features make the most beneficial auxiliary tasks. The content for this chapter is drawn from Benton, Mitchell, and Hovy (2017), a long paper in EACL 2017, and the majority of experiments were performed as part of the 2016 JSALT workshop.

5.1 Motivation

Suicide is one of the leading causes of death worldwide, and over 90% of individuals who die by suicide experience mental health conditions.[1] However, detecting the risk of suicide, as well as monitoring the effects of related mental health conditions, is challenging. Traditional methods rely on both self-reports and impressions formed during short sessions with a clinical expert, but it is often unclear when suicide is a risk in particular.[2] Consequently, conditions leading to preventable suicides are often not adequately addressed.

Automated monitoring and risk assessment of patients' language has the potential to complement traditional assessment methods, providing objective measurements to motivate further care and additional support for people with difficulties related to

[1] https://www.nami.org/Learn-More/Mental-Health-Conditions/Related-Conditions/Suicide#sthash.dMAhrKTU.dpuf

[2] Communication with clinicians at the 2016 JSALT workshop (Hollingshead, 2016).

mental health. This paves the way toward verifying the need for additional care with insurance coverage, for example, as well as offering direct benefits to clinicians and patients.

We explore some of these possibilities in the mental health space using *written social media text* that people with different mental health conditions are already producing. Uncovering methods that work with such text provides the opportunity to help people with different mental health conditions by leveraging a data source they are already contributing to.

Social media text carries implicit information about the author, which has been modeled in natural language processing (NLP) to predict author characteristics such as *age* (Goswami, Sarkar, and Rustagi, 2009; Rosenthal and McKeown, 2011; Nguyen et al., 2014), *gender* (Sarawgi, Gajulapalli, and Choi, 2011; Ciot, Sonderegger, and Ruths, 2013; Liu and Ruths, 2013; Volkova et al., 2015b; Hovy, 2015), *personality* (Schwartz et al., 2013b; Volkova, Coppersmith, and Van Durme, 2014a; Plank and Hovy, 2015; Park et al., 2015; Preoţiuc-Pietro et al., 2015), and *occupation* (Preotiuc-Pietro, Lampos, and Aletras, 2015). Similar text signals have been effectively used to predict mental health conditions such as *depression* (De Choudhury et al., 2013; Coppersmith et al., 2015a; Schwartz et al., 2014), *suicidal ideation* (Coppersmith et al., 2016; Huang et al., 2015), *schizophrenia* (Mitchell, Hollingshead, and Coppersmith, 2015) or *post-traumatic stress disorder (PTSD)* (Pedersen, 2015).

However, these studies typically model each condition in isolation, which misses the opportunity to model coinciding influence factors. Tasks with underlying commonalities (e.g., part-of-speech tagging, parsing, and NER) have been shown to benefit from multi-task learning (MTL), as the learning implicitly leverages interactions between them (Caruana, 1993; Sutton, McCallum, and Rohanimanesh, 2007; Rush

et al., 2010; Collobert et al., 2011; Søgaard and Goldberg, 2016). Suicide risk and related mental health conditions are therefore good candidates for modeling in a multi-task framework.

In this chapter we apply multi-task learning for detecting suicide risk and mental health conditions. The tasks in our model include the user mental health conditions of *neuroatypicality* (i.e. having an atypical mental condition) and *suicide attempt*, as well as the related mental health conditions of *anxiety*, *depression*, *eating disorder*, *panic attacks*, *schizophrenia*, *bipolar disorder*, and *post-traumatic stress disorder (PTSD)*, and we explore the effect of task selection on model performance. We additionally include the effect of modeling a user demographic feature, *gender*, which has been shown to improve accuracy in tasks using social media text (Volkova, Wilson, and Yarowsky, 2013; Hovy, 2015).

Predicting suicide risk and several mental health conditions jointly opens the possibility for the model to leverage a shared representation for conditions that frequently occur together, a phenomenon known as *comorbidity*. Further including gender reflects the fact that gender differences are found in the patterns of mental health (WHO, 2016), which may help to sharpen the model. The MTL framework we propose allows such shared information across predictions and enables the inclusion of several loss functions with a common shared underlying representation. This approach is flexible enough to extend to factors other than the ones shown here, provided suitable data.

We find that choosing tasks that are prerequisites or related to the main task is critical for learning a strong model, similar to findings in Caruana (1996). We further find that including gender as an auxiliary task improves accuracy across a variety of conditions, including suicide risk. The best-performing model from our experiments demonstrates that multi-task learning is a promising new direction in

automated assessment of mental health and suicide risk, with possible application to the clinical domain.

5.1.1 Findings

1. We demonstrate the utility of MTL in predicting mental health conditions from social user text – a notoriously difficult task (Coppersmith et al., 2015b; Coppersmith et al., 2015a) – with potential application to detecting suicide risk.
2. We explore the influence of task selection on prediction performance, including the effect of gender.
3. We show how to model tasks with a large number of positive examples to improve the prediction accuracy of tasks with a small number of positive examples.
4. We compare the MTL model against a single-task model with the same number of parameters, which directly evaluates the multi-task learning approach.
5. The proposed MTL model increases the True Positive Rate at 10% false alarms by up to 9.7% absolute (for anxiety), a result with direct impact for clinical applications.

5.2 Model Architecture

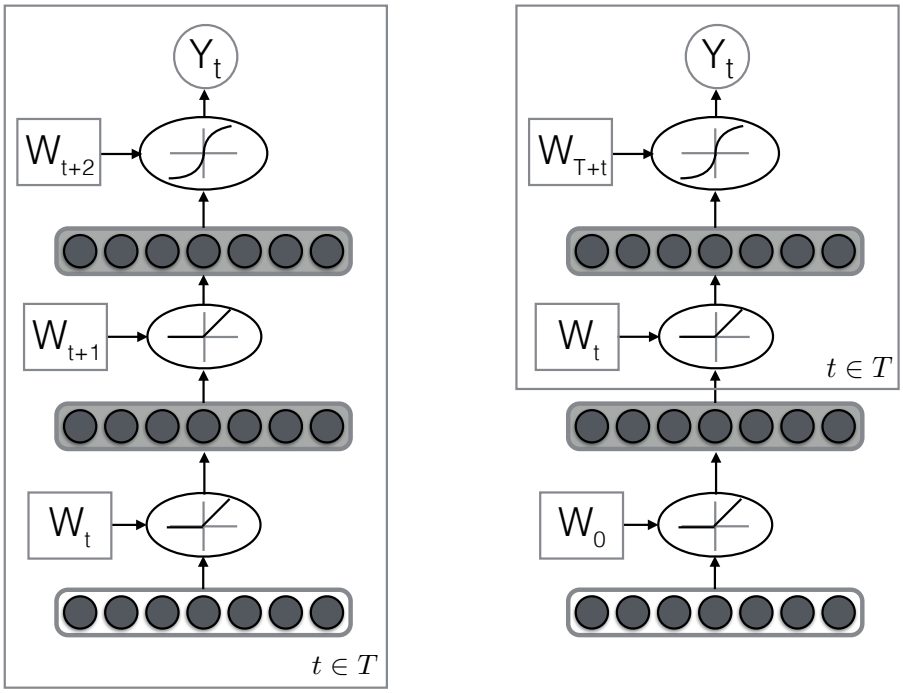
A neural multi-task architecture opens the possibility of leveraging commonalities and differences between mental conditions. Previous work (Collobert et al., 2011; Caruana, 1996; Caruana, 1993) has indicated that such an architecture allows for sharing parameters across tasks, and can be beneficial when there is varying degrees of annotation across tasks. This makes MTL particularly compelling in light of mental

health comorbidity, and given that different conditions have different amounts of associated data.

Previous MTL approaches have shown considerable improvements over single task models, and the arguments are convincing: predicting multiple related tasks should allow us to exploit any correlations between the predictions. However, in much of this work, an MTL model is only one possible explanation for improved accuracy. Another more salient factor has frequently been overlooked: The difference in the expressivity of the model class, i.e., neural architectures vs. discriminative or generative models, and critically, differences in the number of parameters for comparable models. Some comparisons might therefore have inadvertently compared apples to oranges.

In the interest of examining the effect of MTL specifically, we compare the multi-task predictions to models with equal expressivity. We evaluate the performance of a standard logistic regression model (a standard approach to text-classification problems), a multilayer perceptron single-task learning (STL) model, and a neural MTL model, the latter two with equal numbers of parameters. This ensures a fair comparison by decoupling the unique regularization of MTL from the dimensionality-reduction aspect of deep architectures in general.

The neural models we evaluate come in two forms. The first, depicted in plate notation on the left in Figure 5.1 are the STL models. These are feedforward networks with two hidden layers, trained independently to predict each task. On the right in Figure 5.1 is the MTL model, where the first hidden layer from the bottom is shared between all tasks. An additional per-task hidden layer is used to give the model flexibility to map from the task-agnostic representation to a task-specific one. Each hidden layer uses a rectified linear unit as non-linearity. The output layer uses a logistic non-linearity, since all tasks are binary predictions. The MTL model can



Single-task **Multi-task**

Figure 5.1: STL model in plate notation (left): weights trained independently for each task t (e.g., anxiety, depression) of the T tasks. MTL model (right): shared weights trained jointly for all tasks, with task-specific hidden layers. Curves in ovals represent the type of activation used at each layer (rectified linear unit or sigmoid). Hidden layers are shaded.

easily be extended to a stack of shared hidden layers, allowing for a more complicated mapping from input to shared space.[3]

As noted in Collobert et al. (2011), MTL benefits from mini-batch training, which both allows optimization to jump out of poor local optima, and more stochastic gradient steps in a fixed amount of time (Bottou, 2012). We create mini-batches by sampling uniformly from the users in our data, where each user has some subset of the conditions we are trying to predict, and may or may not be annotated with gender. At each mini-batch gradient step, we update weights for all tasks simultaneously. This not only allows for randomization and faster convergence, it also provides a speed-up over the task selection process reported in earlier work (Collobert et al., 2011).

Another advantage of this setup is that we do not need complete information for every instance: learning can proceed with asynchronous updates, dependent on what the data in each batch has been annotated for, while sharing representations throughout. This effectively learns a joint model with a common representation for several different tasks, allowing the use of several "disjoint" data sets, some with limited annotated instances.

5.3 Data

We train models on a union of multiple Twitter user datasets: 1) users identified as having anxiety, bipolar disorder, depression, panic disorder, eating disorder, PTSD, or schizophrenia (Coppersmith et al., 2015b), 2) those who had attempted suicide (Coppersmith et al., 2015c), and 3) those identified as having either depression or PTSD

[3]We tried training a 4-shared-layer MTL model to predict targets on a separate dataset, but did not see any gains over the standard 1-shared-layer MTL model in our application. Different classification tasks require different selections of neural architecture model depth.

	NEUROTYPICAL	ANXIETY	DEPRESSION	SUICIDE ATTEMPT	EATING	SCHIZOPHRENIA	PANIC	PTSD	BIPOLAR	LABELED MALE	LABELED FEMALE
NEUROTYPICAL	4820									-	-
ANXIETY	0	2407								47	184
DEPRESSION	0	1148	1400							54	158
SUICIDE ATTEMPT	0	45	149	1208						186	532
EATING	0	64	133	45	749					6	85
SCHIZOPHRENIA	0	18	41	2	8	349				2	4
PANIC	0	136	73	4	2	4	263			2	18
PTSD	0	143	96	14	16	14	22	191		8	26
BIPOLAR	0	149	120	22	22	49	14	25	234	10	39

Table 5.1: Frequency and comorbidity across mental health conditions.

from the 2015 Computational Linguistics and Clinical Psychology Workshop shared task (Coppersmith et al., 2015a), along with neurotypical gender-matched controls (Twitter users not identified as having a mental condition). Users were identified as having one of these conditions if they stated explicitly they were diagnosed with this condition on Twitter (verified by a human annotator), and the data was pre-processed to remove direction indications of the condition. Coppersmith et al. (2015c) describes how self-identifying tweets were stripped from the data as a preprocessing step. For a subset of 1,101 users, we also manually-annotate gender. The final dataset contains 9,611 users in total, with an average of 3,521 tweets per user. The number of users with each condition is included in Table 5.1. Users in this joined dataset may be tagged with multiple conditions, thus the counts in this table do not sum to the total number of users.

We use the entire Twitter history of each user as input to the model, and split it into character 1-to-5-grams, which have been shown to generaliize better than words for many Twitter text classification tasks (Mcnamee and Mayfield, 2004; Coppersmith et al., 2015b). For instance, a character n-gram representation of a document is

less sensitive to typographical errors than token n-gram features – although a single mistyped character will yield an entirely different token, the misspelled word will share most of its character unigram features with the correctly spelled word. We compute the relative frequency of the 5,000 most frequent n-gram features for $n \in \{1, 2, 3, 4, 5\}$ in our data, and then feed this as input to all models. This input representation is common to all models, allowing for fair comparison.

5.4 Experiments

Our task is to predict suicide attempt and mental conditions for each of the users in these data. We evaluate three classes of models: baseline logistic regression over character n-gram features (LR), feed-forward multilayer perceptrons trained to predict each task separately (STL), and feed-forward multi-task models trained to predict a set of conditions simultaneously (MTL). We experiment with a feed-forward network against independent logistic regression models as a way to directly test the hypothesis that neural classifiers can improve mental condition prediction, particularly when regularized with MTL.

We also perform ablation experiments to see which subsets of tasks help us learn an MTL model that predicts a particular mental condition best. For all experiments, data were divided into five equal-sized folds, three for training, one for tuning, and one for test (we report performance on this fold).

All our models are implemented in Keras[4] with Theano backend and GPU support. We train the models for a total of up to 15,000 epochs, using mini-batches of 256

[4] `http://keras.io/`

eaxmples each. Training time on all five training folds ranged from one to eight hours on a machine with Tesla K40M.

5.4.1 Evaluation Setup

In clinical settings, we are interested in minimizing the number of false positives, i.e., incorrect diagnoses, which can cause undue stress to the patient. We are thus interested in bounding this quantity. To evaluate the performance, we plot the false positive rate (FPR) against the true positive rate (TPR). This gives us a receiver operating characteristic (ROC) curve, allowing us to inspect the performance of each model on a specific task at any level of FPR.

While the ROC gives us a sense of how well a model performs at a fixed true positive rate, it makes it difficult to compare the individual tasks at a low false positive rate, which is also important for clinical application. We therefore report two more measures: the area under the ROC curve (AUC) and TPR performance at FPR=0.1 (TPR@FPR=0.1). We do not compare our models to a majority baseline model, since this model would achieve an expected AUC of 0.5 for all tasks, and F-score and TPR@FPR=0.1 of 0 for all mental conditions – users exhibiting a condition are the minority, meaning a majority baseline classifier would achieve zero recall.

5.4.2 Optimization and Model Selection

Even in a relatively simple neural model, there are a number of hyperparameters that can (and have to) be tuned to achieve good performance. We perform a line search for every model we use, sweeping over ℓ_2 regularization and hidden layer width. We select the best model based on the development loss. Figure 5.4 shows the

performance on the corresponding test sets (plot smoothed by rolling mean of 10 for visibility).

In our experiments, we sweep over the ℓ_2 regularization constant applied to all weights in $\{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5, 1.0, 5.0, 10.0\}$, and hidden layer width (same for all layers in the network) in $\{16, 32, 64, 128, 256, 512, 1024, 2048\}$. We fix the mini-batch size to 256, and 0.05 dropout rate on the input layer. Choosing a small mini-batch size and the model with lowest development loss helps to account for overfitting.

We train each model for 5,000 iterations, jointly updating all weights in our models. After this initial joint training, we select each task separately, and only update the task-specific layers of weights independently for another 1,000 iterations (selecting the set of weights achieving lowest development loss for each task individually). Weights are updated using mini-batch Adagrad (Duchi, Hazan, and Singer, 2011) – this converges more quickly than other optimization schemes we initially experimented with. We evaluate the tuning loss every 10 epochs, and select the model with the lowest tuning loss.

5.5 Results

Figure 5.2 shows the AUC-score of each model for each task separately, and Figure 5.3 the true positive rate at a low false positive rate of 0.1. Precision-recall curves for model/task are in Figure 5.5. STL is a multilayer perceptron with two hidden layers (with a similar number of parameters as the proposed MTL model). The MTL+gender and MTL models predict all tasks simultaneously, but are only evaluated on the main respective task.

Both AUC and TPR (at FPR=0.1) demonstrate that single-task models do not perform nearly as well as multi-task models or logistic regression. This is likely because the neural networks learned by STL cannot be guided by the inductive bias provided by MTL training. Note, however, that STL and MTL are often perform comparably in terms of F1-score, where false positives and false negatives are equally weighted.

Multi-task suicide predictions reach an AUC of 0.848, and predictions for anxiety and schizophrenia are not far behind (Figure 5.2). Interestingly however, schizophrenia stands out as being the only condition to be best predicted with a single-task model. MTL models show improvements over STL and LR models for predicting suicide, neuroatypicality, depression, anxiety, panic, bipolar disorder, and PTSD. The inclusion of gender in the MTL models leads to direct gains over an LR baseline in predicting anxiety disorders: anxiety, panic, and PTSD.

Figure 5.3 illustrates the *true positive rate* – that is, how many cases of mental health conditions that we correctly predict – given a low *false positive rate* – that is, a low rate of predicting people have mental health conditions when they do not. This is particularly useful in clinical settings, where clinicians seek to minimize over-diagnosing, especially when false positives incur an unnecessary, great treatment and emotional cost. In this setting, MTL leads to the best performance across the board, for all tasks under consideration: neuroatypicality, suicide, depression, anxiety, eating, panic, schizophrenia, bipolar disorder, and PTSD. Including gender in MTL further improves performance for neuroatypicality, suicide, anxiety, schizophrenia, bipolar disorder, and PTSD.

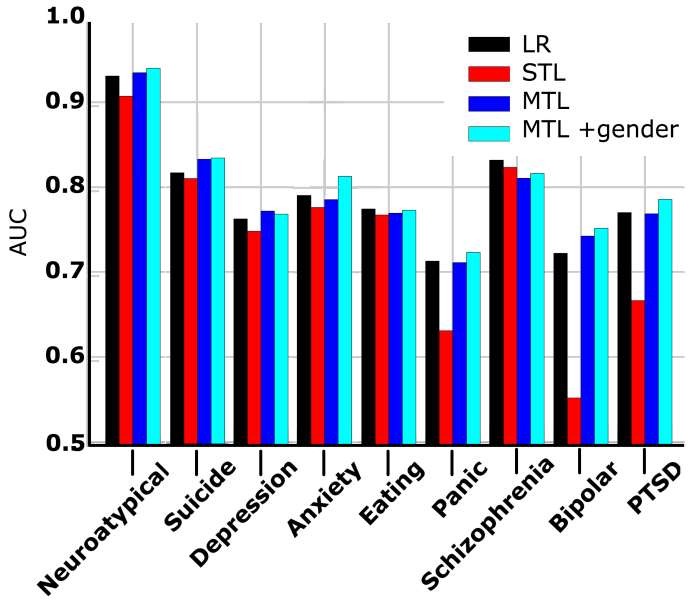


Figure 5.2: AUC for different main mental health prediction tasks.

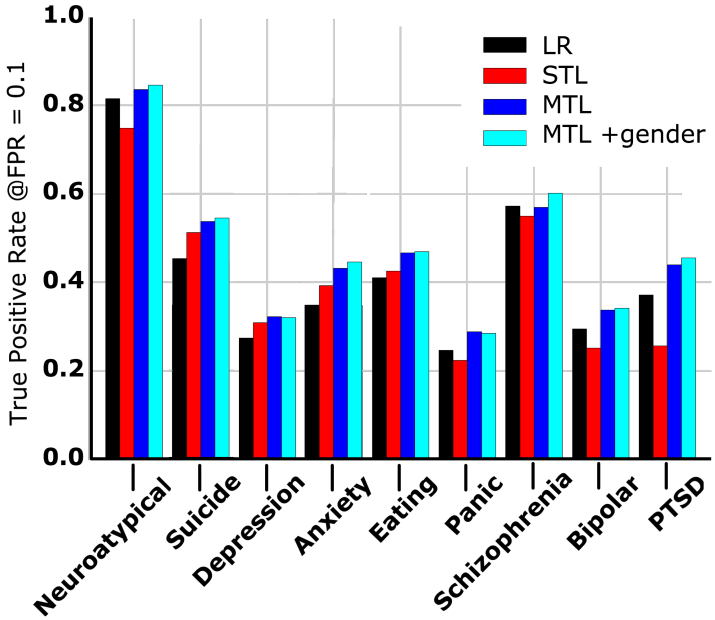


Figure 5.3: TPR at 0.10 FPR for different main mental health prediction tasks.

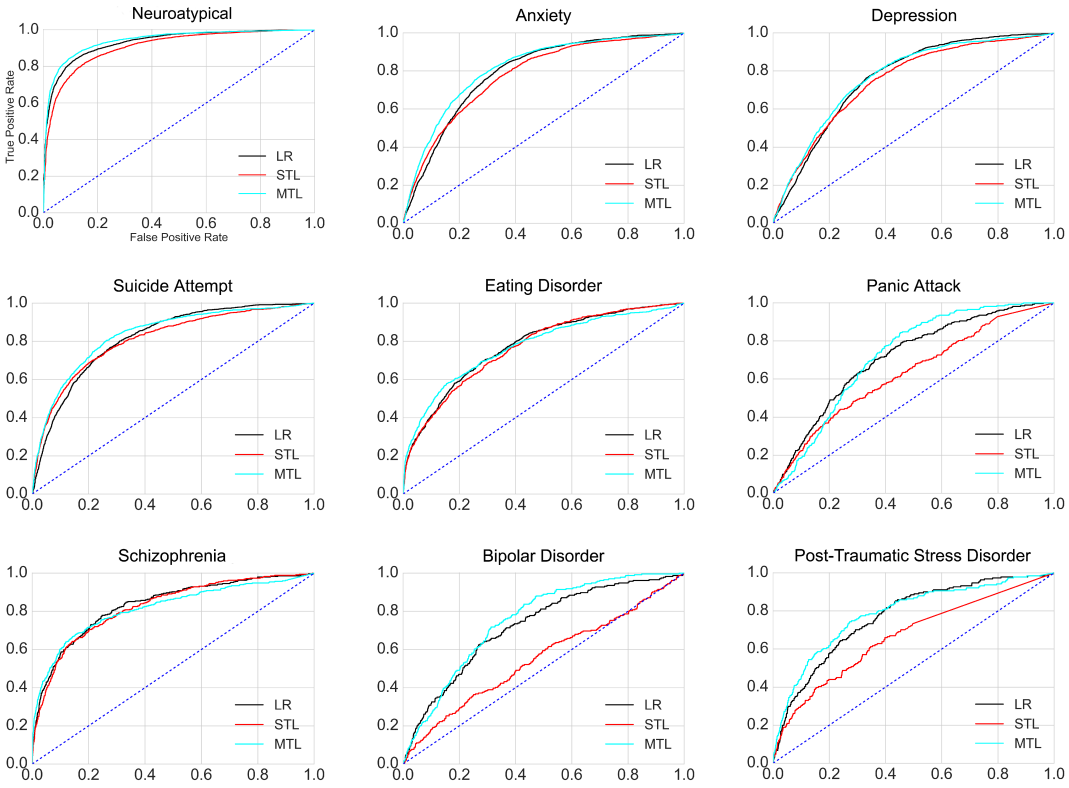


Figure 5.4: ROC curves for predicting each mental health condition. The precision (diagnosed, correctly labeled) is on the y-axis, while the proportion of false alarms (control users mislabeled as having been diagnosed) is on the x-axis. Chance performance is indicated by the blue dotted diagonal line.

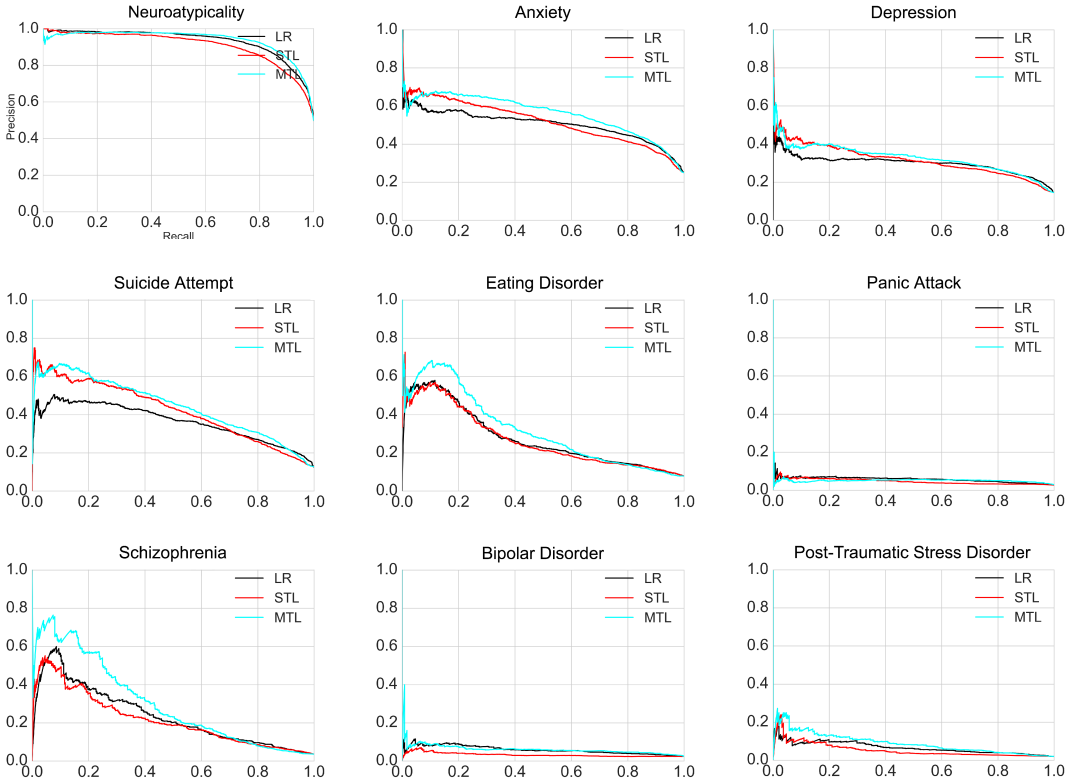


Figure 5.5: Precision-recall curves for predicting each mental health condition.

5.5.1 Comorbid Conditions Improve Prediction Accuracy

We find that the prediction of the conditions with the least amount of data – *bipolar disorder* and *PTSD* – are significantly improved by having the model also predict comorbid conditions with substantially more data: *depression* and *anxiety*. We are able to increase the AUC for predicting PTSD to 0.786 by MTL, from 0.770 by LR, whereas STL fails to perform as well with an AUC of 0.667. Similarly for predicting bipolar disorder (MTL:0.723, LR:0.752, STL:0.552) and panic attack (MTL:0.724, LR:0.713, STL:0.631).

These differences in AUC are significant at $p = 0.05$ according to bootstrap sampling tests with 5,000 samples. The wide difference between MTL and STL can

be explained in part by the increased feature set size – MTL training may, in this case, provide a form of regularization that STL cannot exploit. Further, modeling the common mental health conditions with the most data (depression and anxiety) helps improve performance in predicting rarer conditions comorbid with these common health conditions. This provides evidence that an MTL model can help in predicting elusive conditions by using large data for common conditions, and a small amount of data for more rare conditions.

5.5.2 Utility of Author Demographic Features

Figures 5.2 and 5.3 both suggest that adding an author's demographic feature, such as gender, as an auxiliary task leads to more predictive models, even though the difference is not statistically significant for most tasks. This is consistent with the findings in previous work (Volkova, Wilson, and Yarowsky, 2013; Hovy, 2015). Interestingly, though, the MTL model is worse at predicting gender itself. While this could be a direct result of data sparsity (recall that we have only a small subset annotated for gender), which could be remedied by annotating additional users for gender, this appears unlikely given the other findings of our experiments, where MTL helped in specifically these sparse scenarios.

However, Caruana (1996) notes that not all tasks benefit from a MTL setting in the same way, and that some tasks serve purely auxiliary roles. Here, gender prediction does not benefit from including mental conditions, but guides MTL models to better predict other mental health conditioned. In other words, predicting gender is qualitatively different from predicting mental health conditions: it seems likely that the signals for anxiety are much more similar to the ones for depression than for, say, being male, and can therefore add to detecting depression. However, the distinction

between certain conditions does not add information for the distinction of gender. The effect may also be due to the fact that these data were constructed with inferred gender (used to match controls), so there might be a degree of noise in the data.

5.5.3 Selecting User Features as Auxiliary Tasks

Although MTL tends to dominate STL in our experiments, it is not clear whether modeling several tasks provide a beneficial inductive bias in MTL models in general, or if there exist specific subsets of auxiliary tasks that are most beneficial for predicting suicide risk and related mental health conditions. We perform ablation experiments by training MTL models on a subset of auxiliary tasks, and prediction for a single main task. We focus on four conditions to predict well: suicide attempt, anxiety, depression, and bipolar disorder. For each main task, we vary the auxiliary tasks we train the MTL model with. Since considering all possible subsets of tasks is combinatorially infeasible, we selected the following task subsets as auxiliary:

- *all*: all mental conditions along with gender
- *all conds*: all mental conditions, no gender
- *neuro*: only neurotypicality
- *neuro+mood*: neurotypicality, depression, and bipolar disorder (mood disorders)
- *neuro+anx*: neurotypicality, anxiety, and panic attack (anxiety conditions)
- *neuro+targets*: neurotypicality, anxiety, depression, suicide attempt, bipolar disorder
- *none*: no auxiliary tasks, equivalent to STL

Table 5.2 shows AUC for the four prediction tasks with different subsets of auxiliary tasks. Statistically significant improvements over the respective LR baselines

Auxiliary Tasks	Main Task			
	ANXIETY	BIPOLAR	DEPRESSION	SUICIDE ATTEMPT
all	$0.813^{*\dagger}$	$0.752^{*\dagger}$	$0.769^{\dagger}$	$0.835^{*\dagger}$
all conds	0.786	$0.743^{\dagger}$	$0.772^{\dagger}$	$0.833^{*\dagger}$
neuro	0.763	$0.740^{\dagger}$	0.759	0.797
neuro+mood	0.756	$0.742^{\dagger}$	0.761	0.804
neuro+anx	0.770	$0.744^{\dagger}$	0.746	0.792
neuro+targets	0.750	$0.747^{\dagger}$	0.764	0.817
none (STL)	0.777	0.552	0.749	0.810
LR	0.791	$0.723^{\dagger}$	0.763	0.817

Table 5.2: Test AUC when predicting *Main Task* after multitask training to predict a subset of auxiliary tasks. Significant improvement over LR baseline at $p = 0.05$ is denoted by *, and over no auxiliary tasks (STL) by †.

are denoted by superscript. Restricting the auxiliary tasks to a small subset tends to hurt performance for most tasks, with exception to *bipolar*, which benefits from the prediction of depression and suicide attempt. All main tasks achieve their best performance using the full set of additional tasks as auxiliary. This suggests that the biases induced by predicting different kinds of mental conditions are mutually beneficial – e.g., multi-task models that predict suicide attempt may also be good at predicting anxiety.

Based on these results, we find it useful to think of MTL with user features as a framework to leverage auxiliary tasks as regularization to effectively combat data paucity and less-than-trustworthy labels. As we have demonstrated, this may be particularly useful when predicting mental health conditions and suicide risk.

5.5.4 Discussion

Our results indicate that an MTL framework with user feature tasks can lead to significant gains over single-task models for predicting suicide risk and several mental health conditions. We find benefit from predicting related mental conditions and demographic attributes simultaneously.

We experimented with all the optimizers that Keras provides, and found that Adagrad seems to converge fastest to a good optimum, although all the adaptive learning rate optimizers (such as Adam, etc.) tend to converge quickly. This indicates that the gradient is steeper along certain parameters than others. Default stochastic gradient descent (SGD) was not able to converge as quickly, since it is not able to adaptively scale the learning rate for each parameter in the model – taking too small steps in directions where the gradient is shallow, and too large steps where the gradient is steep. We further note an interesting behavior: all of the adaptive learning rate optimizers yield a strange "step-wise" training loss learning curve, which hits a plateau, but then drops after about 900 iterations, only to hit another plateau. Obviously, we would prefer to have a smooth training loss curve. We can indeed achieve this using SGD, but it takes much longer to converge than, for example, Adagrad. This suggests that a well-tuned SGD would be the best optimizer for this problem, a step that would require some more experimentation and is left for future work.

We also found that feature counts have a pronounced effect on the loss curves: relative feature frequencies yield models that are much easier to train than raw feature counts. This of course is understandable, since feature counts will be sensitive to differences in raw number of tweets between users, whereas relative feature frequencies will be less sensitive.

Learning Rate	Loss	L2	Loss	Hidden Width	Loss
10^{-4}	5.1	10^{-3}	2.8	32	3.0
$5 * 10^{-4}$	2.9	$5 * 10^{-3}$	2.8	64	3.0
10^{-3}	2.9	10^{-2}	2.9	128	2.9
$5 * 10^{-3}$	2.4	$5 * 10^{-2}$	3.1	256	2.9
10^{-2}	2.3	0.1	3.4	512	3.0
$5 * 10^{-2}$	2.2	0.5	4.6	1024	3.0
0.1	20.2	1.0	4.9		

Table 5.3: Average development set loss over epochs 990-1000 of joint training on all tasks as a function of different learning parameters. Models were optimized using Adagrad with hidden layer width 256 (aside for the rightmost column which sweeps over hidden layer width.).

Feature representations are therefore another area of optimization, e.g. different ranges of character n-grams ($n > 5$). We used character 1-to-5-grams, since we believe that these features generalize better to a new domain (e.g., Facebook) than word unigrams. However, there is no fundamental reason not to choose longer character n-grams, other than time constraints in regenerating the data, and accounting for overfitting with proper regularization.

Initialization is a decisive factor in neural models, and Goldberg (2015) recommends repeated restarts with differing initializations to find the optimal model. In an earlier experiment, we tried initializing an MTL model (without task-specific hidden layers) with pretrained word2vec embeddings of unigrams trained on the Google News n-gram corpus. However, we did not notice an improvement in F-score. This could be due to the other factors, though, such as feature sparsity.

Table 5.3 shows parameters sweeps with hidden layer width 256, training the MTL model on the social media data with character trigrams as input features. The sweet spots in this table may be good starting points for training models in future work.

5.5.5 Related Work

Some of the first works on MTL were motivated by medical risk prediction (Caruana, Baluja, and Mitchell, 1996), and it is now being rediscovered for this purpose (Lipton et al., 2016). The latter use a long short-term memory (LSTM) structure to provide several medical diagnoses from health care features (yet no textual or demographic information), and find small, but probably not significant improvements over a structure similar to the STL we use here.

The target in previous work was medical conditions as detected in patient records, not mental health conditions in social text. The focus in this work has been on the possibility of predicting suicide attempt and other mental health conditions using social media text that a patient may already be writing, without requiring full diagnoses.

The framework proposed by Collobert et al. (2011) allows for predicting any number of NLP tasks from a convolutional neural network (CNN) representation of the input text. The model we present is much simpler: A feed-forward network with n-gram input layer, and we demonstrate how to constrain n-gram embeddings for clinical application. Comparing with additional model architectures is possible, but distracts from the question of whether MTL training with user features can improve mental condition prediction in this domain. As we have shown, it can.

5.6 Summary

In this chapter we showed that user mental health and gender features can be used to learn more accurate suicide risk and mental health classifiers from Twitter user text. Integrating user features as auxiliary tasks during training is clearly a more effective way to integrate user features into a classifier than treating them as predictors, since

properties like user mental condition and gender are not available at test time. This shows that user features can improve machine learning models that broadly improve public health.

Our results show that an MTL model trained to predict all user mental health tasks performs significantly better than other models, reaching 0.846 true positive rate for predicting neuroatypicality at a false positive rate of 0.1 (TPR@FPR=0.1), and a TPR@FPR=0.1 of 0.559 for predicting suicide risk. Due to the nature of MTL, we also find pronounced gains in detecting anxiety, PTSD, and bipolar disorder. MTL predictions for anxiety, for example, reduce the error rate from a single-task model by up to 11.9%.

Our results also underscore the general challenge neural models face in defeating strong linear models with scarce training data. Logistic regression classifiers predict a single mental condition more accurately than feedforward neural networks trained on a single task. It is only with the beneficial regularization of user demographic and mental condition tasks and that neural networks outperform logistic regression. This suggests that explicitly designing a neural architecture with the classification task in mind can make the critical difference between under or overperforming a baseline linear model. In this case, an architecture of a "forest" of tasks corresponding to correlated user demographic and mental condition comorbidities improved mental condition prediction.

Whether user embeddings can act as useful auxiliary tasks for learning mental health classifiers is still open. However, they may be noisy surrogates for user gender, age, and other demographic features, as evidenced by the experiments in Section 3.5.3. Therefore, it is natural to assume that user embeddings would be useful auxiliary targets in cases where predicting user demographic properties are related to the main

task. Chapter 6 follows this line of research by exploring whether user embeddings are beneficial auxiliary tasks in an MTL framework to improve tweet-level stance classifiers.

Chapter 6

User Embeddings to Improve Tweet Stance Classification

Chapter 5 showed that ground truth user features – mental condition and demographic features – help learn more accurate classifiers, specifically at predicting the mental conditions a Twitter user has based on their character n-gram usage in tweets they post. This was accomplished by training neural classifiers in a MTL framework where additional user conditions and gender were added as auxiliary tasks. The question remains: can user embeddings take the place of ground truth user features and also act as beneficial auxiliary tasks? This chapter answers this question (in the affirmative!) for the domain of tweet stance classification. This is more evidence that semi-supervised training, predicting user embeddings as an initial auxiliary task, can be used to improve a wide range of tasks beyond predicting latent user features.

In this chapter we consider recurrent neural network (RNN) tweet-level stance classifiers, and evaluate the efficacy different pretraining schemes. We evaluate on two separate datasets: 1) the hashtag-annotated Twitter gun control opinion dataset described in Chapter 4 and 2) the stance classification dataset released as part of the SemEval 2016 6A shared task. We show that user embeddings alone are surprisingly

effective at predicting gun control stance. We then proceed to use the author embeddings indirectly, as auxiliary tasks to pretrain the parameter-heavy RNN stance classifiers. We find that this pretraining improves stance classification performance on average across the five domains in the SemEval shared task, although it still performs on par or underperforms compared to linear classifiers trained on tweet token n-gram features.

Section 6.1 introduces the problem of stance classification. Section 6.4 then describes the different datasets used for training stance classifiers as well as learning the user embeddings used in pretraining. Section 6.5 discusses the experimental setting and section 6.3 describes the model architectures we evaluate in detail. Finally, section 6.6 presents performance of user embeddings for predicting stance alone, along with the performance of RNNs. This work was presented at W-NUT 2018 (Benton and Dredze, 2018b).

6.1 Introduction

Social media analyses often rely on a tweet classification step to produce structured data for analysis, including tasks such as sentiment (Jiang et al., 2011) and stance (Mohammad et al., 2016) classification. Common approaches feed the text of each message to a classifier, which predicts a label based on the content of the tweet. However, many of these tasks benefit from knowledge about the context of the message, especially since short messages can be difficult to understand (Aramaki, Maskawa, and Morita, 2011; Collier and Doan, 2011; Kwok and Wang, 2013). One of the best sources of context is the message author herself. Consider the task of stance classification, where a system must identify the stance towards a topic expressed in a

tweet. Having access to the latent beliefs of the tweet's author would provide a strong prior as to their expressed stance, e.g. general political leanings provide a prior for their statement on a divisive political issue. Therefore, we propose providing user level information to classification systems to improve classification accuracy.

One of the challenges with accessing this type of information on social media users, and Twitter users in particular, is that it is not provided by the platform. While political leanings may be helpful, they are not directly contained in metadata or user-provided information. Furthermore, it is unclear which categories of user information will best inform each classification task. While information about the user may be helpful in general, *what* information is relevant to each task may be unknown.

We propose pretraining tweet stance classifiers to predict a user embedding given the tweet text. This is similar to multitask training of mental health classifiers in Chapter 5, where ground truth binary user features were used as auxiliary tasks, instead of embeddings. Since a deployed classifier will likely encounter many new users for which we do not have embeddings, we use the user embeddings as a mechanism for pretraining the classification model By pretraining model weights to be predictive of user embeddings, a classifier will be able to generalize better on heldout data after training on a task-specific dataset. This pretraining can be performed on a separate, unlabeled dataset of tweets and user embeddings and tends to improve downstream task performance. Although semi-supervised approaches to stance classification are far from new, they have been implemented at the message-level – predicting heldout hashtags from a tweet, for example (Zarrella and Marsh, 2016). Our approach leverages additional user information that may not be contained in a single message.

In this chapter, we evaluate our approach on two stance classification datasets: 1) the SemEval 2016 task of stance classification (Mohammad et al., 2016) and 2) the

guns-related Twitter opinion data described in Section 4.4.2. On both datasets we compare the benefit of pretraining neural stance classifiers to predict different user embeddings derived from different types of online user activity: an author's ego text user embedding, their friend network embedding, and a multiview embedding of both of these views. We also compare pretraining on within-domain user embeddings vs. pretraining on the generic out-of-domain user embeddings learned in Chapter 3.

6.2 Stance Classification

The popularity of sentiment classification is motivated in part by the utility of understanding the opinions expressed by a large population (Pang and Lee, 2008). Sentiment analysis of movie reviews (Pang, Lee, and Vaithyanathan, 2002) can produce overall ratings for a film, analysis of product reviews allow for better recommendations (Blitzer, Dredze, and Pereira, 2007), and analysis of opinions on important issues can serve as a form of public opinion polling (Tumasjan et al., 2010; Bermingham and Smeaton, 2011).

Although similar to sentiment classification, stance classification concerns the identification of an author's position with respect to a given target (Anand et al., 2011; Murakami and Raymond, 2010). This is related to the task of targeted sentiment classification, in which both the sentiment and its target must be identified (Somasundaran and Wiebe, 2009). In the case of stance classification, we are given a fixed target, e.g. a political issue, and want to predict the opinion of a piece of text towards that issue. While stance classification can be expressed as a complex set of opinions and attitudes (Rosenthal, Farra, and Nakov, 2017), we confine ourselves to the task of binary stance classification, in which we seek to determine if a single message expresses support for

or opposition to the given target (or neither). This definition was used in the SemEval 2016 task 6 stance classification task (Mohammad et al., 2016).

A key observation behind stance classification is that the system is designed to uncover the latent position held by the author of the message. While most work in this area seeks to infer the author's position based only on the given message, other information about the author may be available to aid in the analysis of a message. Consider a user who frequently expresses liberal positions on a range of political topics. Even without observing any messages from the user about a specific liberal political candidate, we can reasonably infer that the author would support the candidate. Therefore, when given a message from this author whose target is the political candidate, our model should have a strong prior to predict a positive label.

This type of information is readily available on social media platforms where we can observe multiple messages from a user, as well as other behaviors such as sharing content, liking or promoting content, and the social network around the user. Additionally, this type of contextual information is most needed in a social media setting. Unlike long form text common in sentiment analysis of articles or reviews, analysis of social media messages necessitates understanding short, informal text. Context becomes even more important in a setting that is challenging for NLP algorithms to operate in.

How can we best make use of contextual information about the author? Several challenges present themselves:

First, what contextual information is valuable to social media stance classifiers? We may have previous messages from the user, social network information, and a variety of other types of online behaviors. How can we best summarize a wide array of user behavior in an online platform into a single, concise representation?

We answer this question by exploring several representations of this context encoded a user embedding: a low dimensional representation of the user that can be used as features by the classification system. We include a multiview user embedding that is design to summarize multiple types of user information into a single embedding, learned in Chapter 3.

Second, how can we best use contextual information about the author in the learning process? Ideally we would be provided a learned user representation along with every message we were asked to classify. This is unrealistic. Learning user representations requires data to be collected for each user and computation time to process that data. Neither of these are available in many production settings, where millions of messages are streamed on a given topic. It is impractical to insist that additional information be collected for each user, new representations inferred, all while the consumer of a stance classifier waits for a label to be predicted for a single tweet.

Instead, we integrate user context in multitask learning setting, in a similar way to how user gender was used as an auxiliary task to improve mental condition classification in Chapter 5. We consider augmenting neural models with a pretraining step that updates model weights according to an auxiliary objective function based on available user representations. This pretraining step initializes the hidden layer weights of the stance classification neural network so that the resulting model improves even when observing only a single message at classification time.

Finally, while our focus is stance classification, this approach is applicable to a variety of document classification tasks in which author information can provide important insights in solving the classification problem.

6.3 Models

Our stance classification tasks focus on tweets: short snippets of informal text. We rely on recurrent neural networks as a base classification model, as they have been effective classifiers for this type of data (Tang, Qin, and Liu, 2015; Vosoughi, Vijayaraghavan, and Roy, 2016; Limsopatham and Collier, 2016; Yang et al., 2017).

Our base classification model is a gated recurrent unit (GRU) recurrent neural network classifier. The GRU consumes the input text as a sequence of tokens and produces a sequence of final hidden state activations. Prediction is based on a convex combination of these activations, where the combination weights are determined by global dot-product attention using the final hidden state as the query vector (Luong, Pham, and Manning, 2015). A final softmax output layer predicts the stance class labels based on the convex combination of hidden states. Input layer word embeddings are initialized with GloVe embeddings pretrained on Twitter text (Pennington, Socher, and Manning, 2014).

For this baseline model, the RNN is fit directly to the training set, without any pretraining, i.e. training maximizes the likelihood of class labels given the input tweet. As in Chapters 4 and 5, we have the option of exploring an entire zoo of neural architectures. This is however not the point of this thesis – we want to show how user features and embeddings can be used to improve downstream tasks; indiscriminately exploring different architectures distracts from this point.

We now consider an enhancement to our base model that incorporates user embeddings.

RNN Classifier with User Embedding Pretraining We augment the base RNN classifier with an additional final (output) layer to predict an auxiliary user embedding for the tweet author. The objective function used for training this output layer depends on the type of user embedding (described below). A single epoch is made over the pretraining set before fitting to train.

In this case, the RNN must predict information about the tweet author in the form of an d-dimensional user embedding based on the input tweet text. If certain dimensions of the user embedding correlate with different stances towards the given topic, the RNN will learn representations of the input that predict these dimensions, thereby initializing the RNN with good representations for determining stance.

The primary advantage of this semi-supervised setting is that it decouples the stance classification annotated training set from a set of user embeddings. It is not always possible have a dataset with stance-labeled tweets as well as user embeddings for each tweet author (as is the case for our datasets). Instead, this setting allows us to utilize a stance-annotated corpus, and separately create representations for a disjoint set of pretraining users, even without knowing the identity of the authors of the stance-annotated tweets.

6.3.1 User Embedding Models

We explore pretraining on several different user embeddings. These methods capture both information from previous tweets by the user as well as social network features.

Keyphrases In some settings, we may have a set of important keyphrases that we believe to be correlated with the stance we are trying to predict. Knowing which phrases are most commonly used by an author may indicate the likely stance of that

author to the given issue. We consider how an author has used keyphrases in previous tweets by computing a distribution over keyphrase mentions and treat this distribution as their user representation.

Author Text When a prespecified list of keyphrases is unknown, we include all words in the user representation. Rather than construct a high dimensional embedding – one dimension for each type in the vocabulary – we reduce the dimensionality by applying principal component analysis (PCA) to the TF-IDF-weighted user-word matrix based on tweets from authors (latent semantic analysis) (Deerwester et al., 1990). We use the 30,000 most frequent token types after stopword removal.

Social Network On social media platforms, people friend other users who share common beliefs (Bakshy, Messing, and Adamic, 2015). These beliefs may extend to the target issue in stance classification. Therefore, a friend relationship can inform our priors about the stance held by a user. We construct an embedding based on the social network by creating an adjacency matrix of the 100,000 most frequent Twitter friends in our dataset (users whom the ego user follows). We construct a PCA embedding of the local friend network of the author.

MultiView Representations Finally, we consider a canonical correlation analysis (CCA) multiview embedding over the content of the user's messages as well as their social network[1]. We project both the text and friend network PCA embeddings described above, and take the mean projection of both views as a user's embedding

[1]In actuality, we fit a *GCCA* model to these two views using the `wgcca` library: `https://github.com/abenton/wgcca`. However, as Kettenring (1971) notes, when the number of views is two, this reduces to the same solution as CCA. Thus we refer to it as CCA going forward.

We use a mean squared error loss to pretrain the RNN on these embeddings since they are all real-valued vectors. When pretraining on a user's keyphrase distribution, we instead use a final softmax layer and minimize cross-entropy loss.

For embeddings that rely on content from the author, we collected the most recent 200 tweets posted by these authors using the Twitter REST API[2]. If the user posted fewer than 200 public tweets, then we collected all of their tweets. We constructed the social network by collecting the friends of users as well[3]. We collected user tweets and networks between May 5 and May 11, 2018.

We considered user embedding widths between 10 and 100 dimensions, but selected dimensionality 50 based on an initial grid search to maximize cross validation (CV) performance for the author text PCA embedding.

6.3.2 Baseline Models

We compare our approach against the following two baseline models:

Hashtag Prediction Pretraining As part of the SemEval 2016 task 6 tweet stance classification task, Zarrella and Marsh (2016) submitted an RNN-LSTM classifier that used an auxiliary task of predicting the hashtag distribution *within* a tweet to pretrain their model. There are a few key differences between our proposed method and this work. Their approach is restricted to the stance classification dataset, whereas we consider building representations of the user from context. Additionally, their method is restrictive in that they are predicting a task-specific set of hashtags, whereas user features/embeddings offer more flexibility in that they are not as strongly tied to a specific task. However, we select this as a baseline for comparison because of how

[2] `https://api.twitter.com/1.1/statuses/user_timeline.json`
[3] `https://api.twitter.com/1.1/friends/list.json`

Topic	Count	Hashtags
Atheism	38,667	#jesus, #atheist, #bible, #christ, #god, #lord, #islam, #atheists, #religion, #christian, #christians, #islamophobia, #quran, #christianity, #atheism, #judaism, #allah, #secularism, #sacrilegesunday, #secular, #humanism, #godless, #athiest, #faith, #evolution, #islamicstate, #atheistrollcall, #muhammad, #muslim
Climate Change is a Real Concern	12,417	#cop21, #climatechange, #climate, #globalwarming, #science, #environment, #climateaction, #actonclimate, #paris, #energy, #sustainability, #water, #renewables, #nuclear, #climatechangeisreal, #solar, #parisagreement, #co2, #coal, #cop21paris, #fracking, #action2015, #carbon, #climatemarch, #green, #pollution, #sdgs, #agriculture, #nature, #vegan, #earthtoparis, #geoengineering, #renewableenergy, #junkscience, #keystonexl, #keepitintheground, #oil, #paris2015
Feminist Movement	2,534	#feminismo, #women, #feminism, #feminist, #feminismus, #equality, #fem2, #yesallwomen, #feminisme, #sexism, #womenagainstfeminism, #misogyny, #feminismiscruelty, #gender, #genderequality, #womensrights
Hillary Clinton	734	#demdebate, #gopdebate, #hillary, #hillaryclinton, #stophillary, #whyimnotvotingforhillary
Legalization of Abortion	1,854	#prolife, #abortion, #standwithpp, #waronwomen, #1in3, #defundpp, #ppact
Topic Unclear	19,481	#tcot, #chat, #pjnet, #usa, #trump, #uniteblue, #stoprush, #uk, #philippines, #auspol, #p2, #africa, #australia, #india, #2a, #1a, #nra, #gunfail, #gop, #cdnpoli, #ccot, #makeamericagreatagain, #isis, #truth, #obama, #imwithhuck, #teaparty, #england, #trumptrain, #britain, #health, #tlot, #gamergate, #lgbt, #foodsecurity, #chine, #aus, #arctic, #humanrights, #popefrancis, #blacklivesmatter, #libcrib, #feelthebern, #france, #world, #gunsense, #tntweeters, #london, #politics, #rednationrising, #bernie2016, #tpp, #votetrump2016, #ndp, #berlin, #cruzcrew, #trump2016, #realdonaldtrump, #china, #donaldtrump, #drought, #potus, #parisattacks, #boycott, #c51, #syria, #poverty, #farm365, #chemtrails

Table 6.1: Hashtags used for hashtag prediction pretraining. These were selected based on corpus frequency and hand-curated. They are grouped by topic for presentation, with hashtags that could be relevant to multiple topics in "Topic Unclear". The second column contains the number of times hashtags associated with that topic occurred in the pretraining set.

they utilize hashtags within a tweet for semi-supervised training. We call this model `RNN-content-hashtag`.

We evaluate a similar approach by identifying the 200 most frequent hashtags in the SemEval-hashtag pretraining set (dataset describe below). After removing non-topic hashtags (e.g. #aww, #pic), we were left with 189 unique hashtags, with 32,792 tweets containing at least one of these hashtags (Table 6.1). Pretraining was implemented by using a 189-dimensional softmax output layer to predict held-out hashtag.

RNNs were trained by cross-entropy loss where only the most frequent hashtag was considered to be the target. RNNs were trained by cross-entropy loss where the target distribution placed a weight of 1 on the most frequent hashtag, with all other hashtags having weight of 0. This is the identical training protocol used in Zarrella and Marsh (2016).

There are a few key differences between our proposed method and the MITRE submission. First, the MITRE submission's pretraining regimen relies on predicting tweet-level features, whereas we are predicting user features. Second, their method is restrictive in that they are predicting a task-specific set of hashtags, whereas generic user features or embeddings offer more flexibility in that they are not as strongly tied to a specific task.

SVM Baseline We also reproduce a word and character n-gram linear support vector machine that uses word and character n-gram features. This was the best performing method on average in the 2016 SemEval Task 6 shared task (Mohammad et al., 2016). We swept over the slack variable penalty coefficient to maximize macro-averaged F1-score on held-out CV folds.

6.4 Data

6.4.1 Stance Classification Datasets

We consider two different tweet stance classification datasets, which in total provide six domains of English language Twitter data.

SemEval 2016 Task 6A (Tweet Stance Classification) This is a collection of 2,814 training and 1,249 test set tweets that are about one of five politically-charged targets: *Atheism*, the *Feminist Movement*, *Climate Change is a Real Concern*, *Legalization of Abortion*, or *Hillary Clinton*. Given the text of a tweet and a target, models must classify the tweet as either FAVOR, AGAINST or NEITHER if the tweet does not express support or opposition to the target topic. Many shared task participants struggled with this task, as it was especially difficult due to imbalanced class sizes, small training sets, short examples, and tweets where the target was not explicitly mentioned. See Mohammad et al. (2016) for a thorough description of this data. We report model performance on the provided test set for each topic and perform four-fold CV on the training set for model selection[4].

Guns We built the second stance dataset from the gun control opinion dataset described in Section 4.4.2. Tweets were collected from the Twitter keyword streaming API starting in December 2012 and throughout 2013[5]. The collection includes all tweets containing guns-related keyphrases, subject to rate limits. We labeled tweets based on their stance towards gun control: FAVOR was supportive of gun control, AGAINST was supportive of gun rights. We automatically identified the stance to

[4]CV folds were not released with these data. Since our folds are different than other submissions to the shared task, there are likely differences in model selection.

[5]`https://stream.twitter.com/1.1/statuses/filter.json`

Set Name	Keyphrases/Hashtags
About Guns (General)	gun, guns, second amendment, 2nd amendment, firearm, firearms
Favors	#gunsense, #gunsensepatriot, #votegunsense, #guncontrolnow, #momsdemandaction, #momsdemand, #demandaplan, #nowaynra, #gunskillpeople, #gunviolence, #endgunviolence
Against	#gunrights, #protect2a, #molonlabe, #molonlab, #nogun-control, #progun,#nogunregistry, #votegunrights, #firearm-rights, #gungrab, #gunfriendly

Table 6.2: Keyphrases used to identify gun-related tweets along with hashtag sets used to label a tweet as *Favors* or is *Against* additional gun control legislation.

create labels based on commonly occurring hashtags that were clearly associated with one of these positions (see Table 6.2 for a list of keywords and hashtags). Tweets which contained hashtags from both sets or contained no stance-bearing hashtags were excluded from our data.

We constructed stratified samples from 26,608 labeled tweets in total. Of these, we sampled 50, 100, 500, and 1,000 examples from each class, five times, to construct five small, balanced training sets, and divided the remaining examples equally between development and test sets in each case. We then divided the remaining examples equally between development and test sets in each case. Model performance for each number of examples was macro-averaged over the five training sets. The hashtags used to assign class labels were removed from the training examples as a preprocessing step.

6.4.2 User Embedding Datasets

We considered two unlabeled datasets as a source for constructing user embeddings for model pretraining. Due to data limitations, we were unable to create all of our

Topic	Example hashtags
Atheism	*#nomorereligions, #godswill, #atheism*
Climate Change is a Concern	*#globalwarmingisahoax, #climatechange*
Feminist Movement	*#ineedfeminismbecaus, #feminismisawful, #feminism*
Hillary Clinton	*#gohillary, #whyiamnovotingforhillary, #hillary2016*
Legalization of Abortion	*#prochoice, #praytoendabortion, #plannedparenthood*

Table 6.3: Subset of hashtags used in Mohammad et al. (2016) to identify politically-relevant tweets. We used this set of hashtags to build a pretraining set relevant to the stance classification task.

embedding models for all available datasets. We describe below which embeddings were created for which datasets.

SemEval 2016 Related Users The SemEval stance classification dataset does not contain tweet ids or user ids, so we are unable to determine authors for these messages. Instead, we sought to create a collection of users whose tweets and online behavior would be relevant to the five topics discussed in the SemEval corpus.

We selected query hashtags used in the shared task (Mohammad et al., 2016) and searched for tweets that included these hashtags in a large sample of the Twitter 1% streaming API sample from 2015[6]. Table 6.3 lists the example hashtags described in Mohammad et al. (2016) used to sample politically relevant tweets from the Twitter stream. This ensured that tweets were related to one of the targets in the stance evaluation task, and were from authors discussing these topics in a similar time period. We recorded the author of each of these tweets and then queried the Twitter API to pull the tweet authors' most recent 200 tweets and local friends and followers network. We omitted tweets made by deleted and banned users as well as those who had fewer

[6] `https://stream.twitter.com/1.1/statuses/sample.json`

than 50 tweets total returned by the API. In total, we were able to obtain 79,367 tweets for 49,361 unique users, and were able to pull network information for 38,337 of these users.

For this set of users, we constructed the **Author Text** embedding (PCA representation of a TF-IDF-weighted bag of words from the user) as well as the **Social Network** embedding (PCA representation of the friend adjacency matrix.) For users with missing social network information, we replaced their network embedding with the mean embedding over all other users. This preprocessing was applied before learning **Multiview** embeddings over all users.

General User Tweets Is it necessary for our pretraining set to be topically-related to the stance task we are trying to improve, or can we consider a generic set of users? To answer this question we created a pretraining set of randomly sampled users, of over 102 thousand user learned in Chapter 3, not specifically related to any of our stance classification topics. If these embeddings prove useful, it provides an attractive method whereby general embeddings can be created for users not specifically related to the stance classification topic.

Although there are many potential user embeddings we could consider pretraining with, we only consider the ego text, friend network, and a CCA embedding of these two views as user embeddings for pretraining. We selected these since the PCA ego text embedding is a clear baseline, the friend network embedding was shown to be most effective at friend network prediction, and a CCA representation of ego text and friend network was shown to outperform other embeddings at hashtag prediction. We avoided considering CCA embeddings of all subsets of viwes to narrow the model search space.

To pretrain classifiers, we randomly selected three tweets that each user made in March 2015 as pretraining tweets. Embeddings were learned over tweets from January and February 2015, a disjoint sample from the three randomly selected tweets from March. This resulted in a pretraining set of 152,751 tweets for 61,959 unique users.

Guns User Tweets We also kept 49,023 unlabeled guns tweets for pretraining on the gun control stance task, using the distribution over *general* keyphrases that an author posted across the pretraining set as the user embedding. We pretrained on the (**Author Text**) embedding of these tweets, along with a **Social Network** embedding (network data collected identically to above pretraining datasets).

6.5 Model Training

We preprocessed all tweets by lowercasing and tokenizing with a Twitter-specific tokenizer (Gimpel et al., 2011)[7]. We replaced usernames with <user> and URLs with <url>.

For training on the SemEval dataset, we selected models based on four-fold cross validation macro-averaged F1-score for FAVOR and AGAINST classes (the official evaluation metric for this task). For the gun dataset we select models based on average development set F1-score. For SemEval, each classifier is trained independently for each target. Reported test F1-score is averaged across each model fit on CV folds.

All neural networks were trained by minibatch gradient descent with Adam (Kingma and Ba, 2014) with base step size 0.005, $\beta_1 = 0.99$, and $\beta_2 = 0.999$, with minibatch size of 16 examples, and the weight updates were clipped to have an l2-norm of 1.0. Models were trained for a minimum of 5 epochs with early stopping

[7] https://github.com/myleott/ark-twokenize-py

Hyperparameter	Min	Max
hidden unit width	10	1000
dropout probability	0.0	0.9
word embedding width	25	200
number layers	1	3
directionality	forward	bidirectional

Table 6.4: Grid search range for different architecture and training parameters.

after 3 epochs if held-out loss did not improve, and the loss per example was weighted by the inverse class frequency of the example[8].

The neural model architecture was selected by performing a grid search over hidden layer width ({25, 50, 100, 250, 500, 1000}), dropout rate ({0, 0.1, 0.25, 0.5}), word embedding width ({25, 50, 100, 200}), number of layers ({1, 2, 3}), and RNN directionality (forward or bidirectional). Architecture was selected to maximize cross-fold macro-averaged F1 on the "Feminist Movement" topic with the GRU classifier without pretraining. We performed a separate grid search of architectures for the pretraining models.

Table 6.4 lists the range of hyperparameters swept by the grid search. Figure 6.1 displays average cross-fold F1 for an RNN pretrained on predicting the ego text user embedding during grid search along with on the SemEval 2016 hashtag-filtered pretraining set.

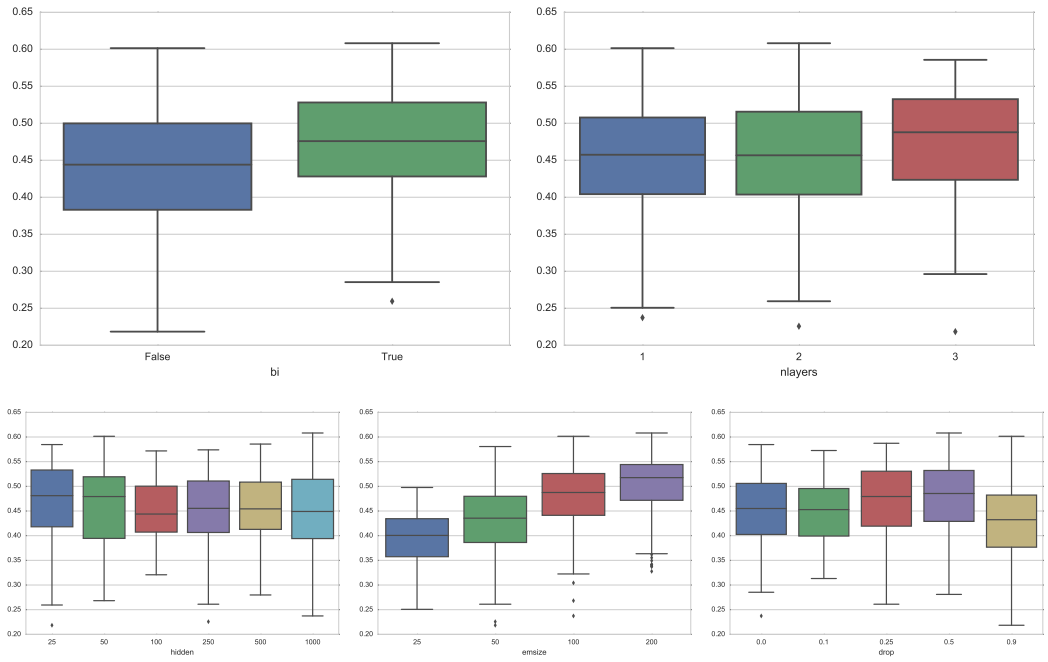


Figure 6.1: Boxplots of mean cross-fold F1-score as a function of different hyperparameters: RNN bi-directionality (upper left), number of layers (upper right), hidden layer width (lower left), embedding width (lower center), and dropout rate (lower right) for an **Author Text**-pretrained RNN on the "Feminism Movement" stance classification task.

Model	Target					
	Ath	Cli	Fem	Hil	Abo	Avg
SVM	61.2	41.4	57.7	52.0	59.1	54.3
RNN	$54.0^{\triangledown}$	39.6	$48.5^{\triangledown}$	**53.5**	58.6	50.8
RNN-content-hashtag	53.4	41.0	$48.4^{\triangledown}$	48.0	55.8	49.3
RNN-hset	58.2	**44.5**	51.2	50.9	**60.2**	**53.0**
RNN-text-hset	58.2	**44.5**	51.2	50.9	**60.2**	**53.0**
RNN-network-hset	42.7	38.8	48.2	42.0	45.0	43.3
RNN-multiview-hset	**60.1**	40.5	49.9	52.5	56.5	51.9
RNN-genset	56.7	41.9	**54.4**	51.7	56.5	52.2
RNN-text-genset	56.7	38.2	**54.4**$^{\diamondsuit\spadesuit}$	51.7	56.5	51.5
RNN-network-genset	54.6	41.4	47.8	50.5	50.6	49.0
RNN-multiview-genset	57.3	41.9	52.1	50.4	54.4	51.2

Table 6.5: Positive/negative class macro-averaged F1 model test performance at SemEval 2016 Task 6A. *hset*: SemEval 2016 hashtag pretrain set, *genset*: general user pretrain set. The best-performing neural model is in bold. The *RNN-genset* and *RNN-hset* rows contain test performance if we select the pretraining embedding type (text, friend, or CCA) according to CV F1 for each domain. The final column is the macro-averaged F1 across all domains. $^{\diamondsuit}$ means model performance is statistically significantly better than a non-pretrained RNN according to a bootstrap sampling test (p=0.05, with 1000 iterations of sample size 250), $^{\triangledown}$ is worse than SVM, and $^{\spadesuit}$ is better than tweet-level hashtag prediction pretraining.

6.6 Results and Discussion

6.6.1 SemEval 2016 Task 6A

Table 6.5 contains the performance for each target in the SemEval 2016 stance classification task.

Considering the pretrained models versus the non-pre-trained RNN, pretraining improves in four out of five targets. Additionally, one of our models always beats the baseline of tweet-level hashtag distribution pre-training (`RNN-content-hashtag`). Notably, while topic specific user embeddings (`hsetpre`) improve over not pretraining in four out of five cases, the generic user embeddings (`genset`) improves in three out of five cases. This suggests that even embeddings for generic users who don't necessarily discuss the topic of interest can have value in model learning.

In terms of embedding type, embeddings built on the author text tended to be best, but results were not clear.

The linear SVM baseline with word and character n-gram features outperforms neural models in two out of five tasks, and perform the best on average. This agrees with the submissions to the SemEval 2016 6A stance classification task, where the baseline SVM model outperformed all submissions on average – several of which were neural models.

6.6.2 Guns

Using the guns dataset, we sought to understand how the amount of training data affected the effectiveness of model pre-training. Table 6.6 show the accuracy for different models at varying amounts of training data. As the amount of training data

[8]This improved performance for tasks with imbalanced class labels such as the "Climate Change" topic.

Model	# Train Examples			
	100	200	1000	2000
SVM	79.2	81.1	85.9	87.4
RNN	72.2$^{\triangledown}$	79.0	**84.0**	85.3
RNN-keyphrase-gunset	73.1$^{\triangledown}$	76.7	83.6	**85.6**
RNN-text-gunset	72.2$^{\triangledown}$	79.0	**84.0**	85.3
RNN-text-genset	71.7$^{\triangledown}$	76.6	83.6	85.3
RNN-network-genset	73.1$^{\triangledown}$	77.2	83.3	85.4
RNN-multiview-genset	**75.0**	**79.1**	83.9	85.4

Table 6.6: Model test accuracy at predicting gun stance. RNNs were pre-trained on either the guns-related pre-training set (*gunset*) or the general user pre-training set (*genset*). The best-performing neural model is bolded. $^{\triangledown}$ indicates that the model performs significantly worse than the SVM baseline ($p <= 0.05$ according to a 1000-fold bootstrap test with sample size 250).

Model	# Train Examples			
	100	200	1000	2000
tweet	**79.2**	**81.1**	85.9	87.4
user-text	72.1$^{\triangledown}$	74.1$^{\triangledown}$	76.5$^{\triangledown}$	76.6$^{\triangledown}$
keyphrase	52.2$^{\triangledown}$	50.8$^{\triangledown}$	51.0$^{\triangledown}$	51.8$^{\triangledown}$
tweet + user-text	**79.2**$^{\clubsuit}$	**81.1**$^{\clubsuit}$	**86.0**$^{\clubsuit}$	**87.6**$^{\clubsuit}$
tweet + keyphrase	**79.2**$^{\clubsuit}$	**81.1**$^{\clubsuit}$	85.9$^{\clubsuit}$	87.4$^{\clubsuit}$

Table 6.7: Test accuracy of an SVM at predicting gun control stance based on guns-related keyphrase distribution (*keyphrase*), user's **Author Text** embedding (*text*), and word and character n-gram features (*tweet*). $^{\triangledown}$ encodes models significantly worse ($p = 0.05$) than a tweet features-only SVM according to a bootstrap sampling test with sample size 250 and 1000 iterations, and $^{\clubsuit}$ means the feature set did significantly better than user-text-PCA.

increases, so does model accuracy. Additionally, we tend to see larger increases from pre-training with less training data overall. It is unclear which user embedding or pre-training set is most effective. The CCA embedding is most effective at improving the neural classifier, although the difference is not statistically significant. The difference may be related to the finding in Section 3.4.2: that multiview embeddings are more predictive of a person's future hashtag use than considering a single view.

As with SemEval, the SVM always outperforms neural models, though the improvement is only statistically significant in the smallest data setting. Although we are unable to beat SVM models, the improvements we observe in RNN performance after user embedding pre-training are promising. Neural model architectures offer more flexibility than SVMs, particularly linear-kernel, and we only consider a single model class (recurrent networks with GRU hidden unit). Further architecture exploration is necessary, and user embedding pre-training will hopefully play a role in training state-of-the-art stance classification models.

Since for the guns data we have an intersection between the annotated stance data and the users for whom we have embeddings, we sought to understand how much information may be contained in the embeddings relevant to stance classification. Like above, we trained an SVM to predict the gun stance but instead of providing the tweet, we alternately provided the tweet, one of the embeddings, or both together. Higher prediction accuracy indicates that the input is more relevant, and helpful, in predicting stance.

Table 6.7 shows test accuracy for this task across different amounts of training data. Unsurprisingly, the tweet content is more informative at predicting stance than the user embedding. However, the embeddings did quite well, with the "Author Text"

embedding coming close to performance of tweet text in some cases. Providing both had no or a marginal improvement over tweet text alone.

6.7 Summary

This chapter shows that pretraining on unsupervised user embeddings improves tweet-level neural stance classifiers. We find that author embedding pretraining yields improvements on four out of five domains for the SemEval 2016 Task 6A tweet stance classification task over a non-pretrained neural network, although benefits are less discernible on the gun control stance classification dataset of tweets.

We expand on Chapter 5 and show that pretraining to predict unsupervised user embeddings also improves classifier performance, in spite of not having gold user features. *This remains true even when pretraining on a completely generic set of user embeddings, when the domains of the training and unlabeled sets do not match.* This suggests that the set of user embeddings compiled in Chapter 3 can be used as a generic target to initialize tweet-level classifiers in multiple domains. This chapter also successfully applies user feature semi-supervised training to a much more modern and slippery collection of neural bells and whistles, a RNN with GRU layers, and an attention mechanism across intermediate hidden states. Compared to the multilayer perceptrons trained in the previous chapter, these stance classifiers are grotesquely baroque.

In Section 3.4.2 we found that multiview embeddings were more predictive of future hashtag use than other combinations of user views. This difference in performance on the SemEval shared task between pretraining on (**Author Text**, **Social Network**) CCA user embedding features vs. **Author Text** alone is not statistically

significant, although pretraining on **Author Text** performs better on average. It is hard to make strong claims on the merits of different embeddings since the stance classification datasets have small evaluation sets.

However, it is clear that pretraining RNN stance classifiers to predict a user's local friend network embeddings actively hurts final performance – **Social Network** embeddings provide a malicious inductive bias, at least for stance classification. This highlights the findings in chapter 3, that different user embeddings are best suited for different downstream tasks. Local friend network embeddings can predict other friending behavior accurately, but they are less effective at informing one's stance on politically-sensitive issues.

Chapter 7

Conclusion

This thesis explores representation learning techniques to learn social media user embeddings and evaluates embeddings at improving downstream task performance. In the process we develop novel methods to learn user embeddings and integrate them into existing models. We conclude by summarizing the contributions of each chapter: whether user embeddings are being learned (and if so how), how they are evaluated, the tasks we try to improve with them, and any novel models described there. In section 7.1 we retrospectively summarize the main contributions of this thesis. In section 7.2 we touch on ethical considerations around social media data and the choices we made in our research to respect the privacy of users. We conclude in section 7.3 with directions for future research.

Chapter 2 begins by reviewing work on applications of user features and then provides an overview of relevant computational methods: canonical correlation analysis (CCA)-based multiview representation learning methods, and the multitask learning setting. Section 2.1 is a review of work on inferring user demographic features and applications that benefit from user features and embeddings. Section 2.2 is an overview of correlation-based multiview representation learning methods covering how these

models are fit and the data assumptions that they make. This section presents the derivation of the vanilla two-view CCA solution, the derivation of many-view generalized CCA, and describes existing extensions to these methods. This section could stand as a primer on correlation-based multiview representation learning. Section 2.3 describes the multitask learning setting, the motivation behind this framework, and describes (at a high level) how learn neural models are learned in this setting.

Chapter 3 describes how a set of multiview embeddings were learned for a general collection of over 100,000 English-speaking Twitter users. There are two main contributions in this chapter.

First, this chapter contains the methodological meat of how multiview user embeddings are learned. An extension of MAXVAR-GCCA is presented. Weighted GCCA (*wGCCA*) introduces scalar weights into the per-view GCCA loss which adjust the penalty placed on failing to recover the latent embedding from features of a particular view. The *wGCCA* per-view weights are tuned by the practitioner based on their belief of which views should be best captured by the user embedding or explicitly tuned for downstream task loss.

Second, this chapter includes experiments on using GCCA methods to learn user embeddings from different views of Twitter user behavior/features including the text they post, who they friend, who follows them, who they mention, and what sort of language their local network uses. The key takeaways from these analyses are: (1) multiview user embeddings learned on ego text along with friending behavior are more predictive of hashtag usage (and presumably word/topic usage) than simple combinations of each of these views, (2) multiview user embeddings are not a panacea for every task, and friend prediction is best captured solely by other friending behavior

and (3) *dGCCA* embeddings considering all user behavior views simultaneously, without discriminatively tuning the loss weighting for each view can predict future hashtag usage better than a linear *wGCCA* embedding with freedom to weight each view. This suggests that relying on nonlinear techniques to learn user embeddings is important, at least when predicting hashtag usage.

Chapter 4 centers on an application of using inferred user location features as supervision for topic models to improve topic model fit on social media text. Topic models are fit to three keyword-filtered Twitter message corpora. We present a new topic model, Deep Dirichlet Multinomial Regression (*dDMR*), which is better-suited to high-dimensional and noisy document supervision than *DMR*. *dDMR* is evaluated on a synthetic dataset as well as a corpus of New York Times articles, Amazon product reviews, and Reddit posts. Although this is presented as an application of using user embeddings to improve topic modeling of social media documents, the neural representation layer of *dDMR* can be viewed as *learning* user embeddings (when using author-specific features are provided). These user embeddings are explicitly trained to be predictive of topic preference.

Chapter 5 describes experiments on predicting suicide risk and mental health conditions for Twitter users given their tweet history. Here we show that training classifiers in a multitask learning framework, with a user's other mental health conditions and gender as auxiliary tasks, learns stronger mental health condition classifiers from Twitter user text than training classifiers independently. This chapter presents the following findings: (1) multitask training for a user's mental health improves performance over independently trained systems where auxiliary tasks are considered other

possible conditions, (2) predicting a demographic feature like gender improves over a model that is only trained to predict mental conditions, and (3) the more auxiliary mental health conditions that are predicted, the stronger the learned classifier. The main contribution is that multitask training of classifiers to predict demographics, and other user features, learns stronger classifiers, better exploiting comorbidities between mental conditions and correlation with demographics.

Chapter 6 describes a similar application as chapter 5, except instead of training a classifier to predict a user's mental condition based on tweet history, we want to train an RNN to predict the stance expressed in a tweet based purely on that tweet's text. We show that on the SemEval 2016 stance classification benchmark, pretraining classifiers to predict user embedding features improves RNN performance. It is even statistically significantly better than pretraining classifiers to predict heldout hashtags from within a tweet. Semi-supervised pretraining is key for the same reason as multitask learning is when predicting user mental health – user features are hard to acquire at test time, but useful when we already have a general collection of user embeddings and tweets to pretrain classifier weights.

7.1 Contributions

This thesis applies multiview representation learning methods to learning user embeddings and evaluates them against other traditional user representations on an array of downstream tasks: improving topic model fit, predicting user demographic features, mental health condition, hashtag usage, friending behavior, as well as improving message-level stance classification on social media. We hope researchers look to these experiments as a guide for determining which user embeddings are most appropriate

for their downstream task and draw inspiration for injecting user embeddings into their own models.

The work presented in this thesis has resulted in three major contributions, as evidenced by subsequently published research.

Expanding What Constitutes as Model Supervision

In chapter 3 we apply several novel variants of multiview representation learning methods for learning social media user embeddings using auxiliary user information as additional views. In chapter 4 we present a new supervised topic model that can exploit high-dimensional, noisy supervision. In chapters 5 and 6 we use neural multitask learning to improve classifier generalization at social media prediction tasks by entraining model weights to also be predictive of features associated with the tweet author.

Although these extensions belong to entirely different model classes (correlation-based multiview learning methods, probabilistic topic models, and supervised neural networks), they were all motivated by the need to exploit the wealth of unstructured, auxiliary information around social media users to improve downstream task performance. This need is not specific to social media data but pervades many applied machine learning domains, encouraging others to arrive at similar solutions. Card, Tan, and Smith (2018) developed a supervised neural topic model that allows for metadata to appear as either a covariate or a predicted variable in the model structure. There is also a line of work that integrates information from multiple user views to learn more robust user embeddings (Li et al., 2017; Tao and Yang, 2017; Kursuncu et al., 2018; Hazarika et al., 2018). The models we present are all trying to extract value out of features that are only distantly related to a task of interest.

By developing these models, we hope to widen the set of viable signals that machine learning practitioners will consider when training semi-supervised models. For example, instead of considering using author demographics as auxiliary tasks when training a multitask model, one may just as well consider predicting prior tweeting history as an auxiliary task (a feature that is readily accessible although higher dimensional).

Learning Social Media User Embeddings

Although representing users as a vector of real numbers is not a new idea, learning user embeddings and evaluating them as first-class objects is a new contribution to social media research. In chapter 3 we evaluate multiview user embeddings at multiple tasks in a similar way to how word embeddings have been subjected to a battery of syntactic and semantic similarity tasks, as well as prediction tasks. Subsequent research has also taken this tact and treats user embeddings as first-class objects worth learning and evaluating in their own right.

For example, Xing and Paul (2017) take inspiration from the friend prediction task to evaluate their own user embeddings. Li et al. (2017) takes a multitask approach to learning user embeddings and evaluates the embeddings according to how well they predict which text and other users they are likely to agree with. Although considering a supervised objective to learn user embeddings, Kursuncu et al. (2018) also collapses features from each view into a joint user embedding. Multiview user embeddings have even been shown to be predictive of sarcasm in author tweets (Hazarika et al., 2018).

State-of-the-art for Social Media Mental Health Monitoring

At the time of publishing, the neural MTL model presented in chapter 5 marked the state-of-the-art for mental health inference from social media text. Tran and Kavuluru (2017) reference this work as follows:

> There is also a quickly growing body of literature detailing machine learned models to predict mental health status based on social media data. For a detailed analysis of the current state-of-the-art in this emerging domain, readers are encouraged to refer to the deep learning architecture by Benton et al. [6].

Subsequent work has also taken semi-supervised approaches to monitor mental health from users' social media data (Yazdavar et al., 2017; Zou, Lampos, and Cox, 2018). In particular, our finding that *classification tasks with little labeled training data tend to benefit more from MTL than tasks with more training data* is frequently cited as justification for the use of MTL (Bingel and Søgaard, 2017) across many application domains: news headline popularity prediction (Hardt, Hovy, and Lamprinidis, 2018), information retrieval (Salehi et al., 2018), medical concept normalization and recognition (Crichton et al., 2017; Niu et al., 2018), and NLP (Bjerva, 2017; Schulz et al., 2018).

7.2 Ethical Considerations

The work described in this thesis can lead to more robust social media systems, benefiting the users whose data these models were tuned to. At the same time, it

is important to recognize that although these analyses were purely observational[1], the users' whose data comprise our training sets may not be comfortable with being studied. The survival of social media research depends on the trust of the users whose data are studied and betraying this trust will can have far-reaching effects[2].

There is a wide set of ethical concerns associated with doing social media research, especially when dealing with users' physical and mental health. These concerns are extensively covered by McKee (2013), Conway (2014), and Ayers et al. (2018). In Benton, Coppersmith, and Dredze (2017) we condense these into a set of maxims and describe ethical concerns that new social media health researchers should be aware of.

In this thesis we made several explicit choices to respect the privacy of users in our studies. In Chapter 3 we chose to only present anecdotes of users that were obfuscated. The user clusters in Appendix A do not include user IDs of cluster members, only a bag of words associated with exemplar users in that cluster. We also made sure not to release tweets made by these users as this would explicitly break the Twitter REST API's terms of service (although we did release the pre-trained user embeddings).

The work in Chapter 5 is the most ethically treacherous. In other chapters, user data is used to improve models that are only tangentially related to these same users, or are used in innocuous tasks (e.g. improving topic model fit, predicting hashtag use, and friending behavior). Although the users in the mental health prediction work made their accounts publicly available, they probably did not expect to be enrolled in an observational study. The stigmatism surrounding mental illness prevented us from sharing their data or identities with outside researchers. We also explicitly chose to

[1] We did not directly engage with the users whose data we collected or attempt to influence their behavior.

[2] https://www.nytimes.com/2018/04/10/us/politics/zuckerberg-facebook-senate-hearing.html

not release the models we trained over their tweets, nor do we release an analysis of which features were most influential in predicting mental health conditions for fear that these character/word choice features would be used to stigmatize others.

7.3 Directions for Future Research

7.3.1 User Embedding Evaluation Suite

Creating a benchmark evaluation suite set for user embeddings similar to word similarity benchmarks for word embeddings (Rubenstein and Goodenough, 1965; Miller and Charles, 1991; Finkelstein et al., 2001; Hill, Reichart, and Korhonen, 2015) would be a valuable addition to systematically comparing user embeddings. Just like word embeddings, user embeddings can always be finetuned for a specific downstream task, but it is not clear whether specific representation learning methods or data sources are a better starting point for a wide array of tasks. The evaluation in chapter 3 is a first step at constructing such a benchmark suite: future hashtag, friend link, and demographic prediction. However, there is a night inexhaustible list of dimensions in which users can be similar to each other: according to socioeconomic status, geographical origin, entertainment preference, personality profiles, etc. A suite that covers a diverse set of user properties would better judge how useful a user embedding is.

7.3.2 Scalable Multiview Representation Learning

Scaling Over Examples: The MV-LSA approximate solution to MAXVAR-GCCA presented in chapter 3 is a batch method that is difficult to scale up to very large datasets. The barrier to scalability arises from ensuring that orthonormality constraints on the columns of G are satisfied (columns span all training examples). Finding a

GCCA embedding for billions of Facebook users would be intractable under this algorithm. Algorithms that solve CCA by updating projections based on a single example at a time such as stochastic AppGrad (Ma, Lu, and Foster, 2015) or stochastic CCA (Arora et al., 2017) do not run into memory constraints as the number of examples increases, but in practice may take far too long to converge to a sufficiently low los solution. Algorithms to scalably solve generalized CCA problems are less mature than those for two-view CCA although there has been some work. For example, Fu et al. (2016) gives an alternating optimization algorithm for the SUMCOR-GCCA problem that decomposes the optimization as a sequence of linear least-squares problems (the *LasCCA* algorithm described in Section 3.3.2), and Fu et al. ("Scalable and Flexible Multiview MAX-VAR Canonical Correlation Analysis") gives a similar algorithm for approximately solving a regularized variant of MAXVAR-GCCA.

Neural architectures have also been proposed to maximizing correlation between more than two views, but these architectures make additional assumptions on which correlations are maximized between views. For example, the Bridge Correlational Networks proposed in Rajendran et al. (2015) assume that one of the views is designated a "pivot" view whose representation all other views are mapped close to. On the other hand, generalized CCA formulations such as SUMCORR-GCCA and MAXVAR-GCCA maximize a loss function dependent on correlation between all pairs of views. Neural proposals also do not come with theoretical guarantees on solution quality, and therefore may have trouble difficulty finding good user embedfdings in practice.

Accounting for Noisy Views: Certain example views will have higher variance than others. Imagine a user has only posted a message once in their online life, but they have friended over 1000 other users. The ego text view for that user will be a

noisier estimate of their true ego text distribution (when they have produced infinite tweets) than their friend network view if they explicitly took the time to vet every other user in the entire network as a potential friend. Although the example-view binary mask in MV-LSA, K, explicitly ignores example views that are missing data, it cannot tell how much to *trust* each view. Rastogi, Van Durme, and Arora (2015) suggests a heuristic weighting of example views in the section on "Handling Missing Data". This could be used to downweight views whose estimates are noisier. There has also been work on Bayesian online classifiers of user demographics that gain confidence as more user information arrives, but it is unclear how to apply these models in the multiview representation learning setting (Volkova and Van Durme, 2015)

Scaling to Variable Feature Dimensionality: Another general problem in learning and updating multiview social media user embeddings is the introduction of new social media members and new vocabulary as time goes on. This will result in new features being introduced into the per-view feature vectors. We are not familiar with work on extending the GCCA problem to cases where input feature dimensionality may grow over time. One possible direction is to incorporate new view features into your model by refreshing the embeddings as a batch periodically. This is not a very satisfying solution and is expensive to apply at scale. A second direction would be to look to the nonlinear mappings in *dGCCA* to map new feature elements to the same feature space. Imagine for instance that one of our views is a representation of our local friend network, but we take the mean of friend user description embeddings as our friend network feature vector. Adding a new friend would only require embedding their user description and integrating it into our view average. This is very much an open problem and these solutions are not particularly satisfying.

7.3.3 Interpretability of User Embeddings

There has been work in interpreting the dimensions of textual embeddings (Li et al., 2015). One strength of distributed user embeddings, their ability to encode many user features in a single dense vector, is also their weakness – user embeddings are opaque. This technique could be applied to user embeddings by calculating the distance between an input user to prototypical users (e.g. man, woman, athlete, or a wine-snob). Although this thesis focuses on learning user embeddings to improve downstream tasks, interpretation of these embeddings will be important in analyzing what they capture and convincing engineers that they are worth including in production systems.

Appendix A

User Embedding Clusters

Here we present the labels assigned by Mechanical Turk subjects to user clusters from three different user embeddings. Each cluster was represented by four user exemplars (with one false exemplar/intruder), corresponding to a single Gaussian in a Gaussian mixture model (subsection 3.5.4). Many of the cluster labels are vague, hinting at the difficulty of this task. For each cluster, we present the assigned labels along with tokens from the four exemplar user tweets.

One important thing to be aware of is that different Gaussians capture more users than others. This is likely because we assume all Gaussians have diagonal, but unconstrained covariance matrices – a single Gaussian can have very large variance and capture many users that do not fit neatly into other, narrower Gaussians. The *Number Members* column in the tables below contains the number of users belonging to each cluster (assigned highest probability under mixture model). Next to each label is a sign for whether the subject correctly identified the intruder ($+$ for correct, $-$ for incorrect).

PCA on Ego Text

Cluster Index	Number Members	Labels	User Text
1	1148	*Tweeting personal thoughts and opinions* (-) *user 5 does not belong tweeting snoop dogg* (-) *different language* (-)	*snap chat instagram bitch bitch guilty guilty always following back toke*
2	960	*other four are women* (-) *The other 4 are female* (-)	*nah tanto como tanto cmo quiera architecture art politics good day three time instagram askfm*
3	374	*Personal accounts talking about their personal lives and interests* (-) *The other four seem less professional* (-)	*film production photographer olympic swimming online editor words stuck mind alive i've learned seize day home returned july*
4	208	*The others seem like homebodies* (+) *It doesn't look like everyone else have kids.* (-)	*proud mommy laila nicole dont fuckery dont ask past wont ask yurs kik #teambi face it's breathing air stop dont followers competition striving better woman yesterday*
5	483	*The other four are young people.* (-) *The other four users are a younger age group* (-) *They are younger people.* (-)	*loving life instagram alexis nothing w/out god love help need forgive hurt pray leave ... romans jesus freak instagram boynton beach florida christian wife mom beautiful girls humbled saving grace jesus*
6	6057	*other four are women* (-) *The other four are young women* (-) *Young women talking about their lives and interests* (-)	*carpe diem reality junkie !!! loves good crafty challenge photography student*

7	7097	*@EvieMarieR is only professional account (Publicist @EdgePublicity.) The rest are personal accounts with no indication about their occupation* (-) *People tweeting about personal, not professional, information* (+)	*baby world everyone else already taken oscar wilde*
8	1183	*Male* (-) *none* (-)	*self proclaimed coolest guy alive also i'm actor comedian new yorker enjoying olvides nunca quien hizo bien pero que ese recuerdo aficionado elche #15 follow*
9	188	*Young people* (+) *NONE* (+)	*make assessment based recent tweets ... don't alphabetical order cats essex fella mine imac ipad iphone kids mine london movies music print publishing thfc travel aspire perspire inspire without god nothing*
10	1374	*English speakers* (+) *none* (-)	*aos del lectora tributo lll amor que mereces ser 0/4 cute bitch every single one 1000 percent boyband tears community activist #hamont survivor realities poverty school hard knocks :-p ;-)*
11	92	*other four are men* (+) *user 5 missing Sagittarius* (+) *They're all men.* (+)	*aint got mind #father #cardnation rip granny cora asshole masters don't give fuck bachelors #teamfollowback #teamfreak #teamokc life short simply organized mess chaos*
12	10316	*Young women talking about their lives* (-) *They are all female twitter users.* (+) *others are female* (+)	*don't get sports physiologist love liverpool sports general #jft96 #ynwa obsessing ugly animals trying act like know i'm*

13	4672	*She's the only non-white person.* (+) *The other 4 feel like they are into american culture* (+)	*like sports stuff*
14	820	*the others are women* (+)	*fans teuku wisnu dan shireen akuntansi uir universitas islam riau i'm adventist accounting university mks 011 september virgo simple #viscabarca*
15	7241	*A politically active group of tweeters* (-)	*welcome lair ... it's filled cosplay cars princesses animals food sexy magic enjoy like like #tgod veteran south african profit organisation committed making human rights real live south africa*
16	81	*The others are men* (-) *The other four are men* (-)	*creative follow get follow back proud directioner fun weird crazy loud chick ..... follow back 100 gunner @arsenal guys follow new twitter yang twitter ini udah ngga pakai thankyou*
17	2823	*A bunch of young Americans tweeting about their lives in English* (+)	*professional movie watcher imitation sincerest form flattery hello*
18	3632	*Communicating interests and personal information* (+) *@petsarefriends is different as Í. display picture of animals while all others have their own picture as DP2̊. most tweets are animal/pet based while all others are varied.* (+)	*dios track&field boca juniors vida entera i'm pet owner lover tattoo artist hard knox tattoos central avenue yonkers check website fewer 160 characters ask*

19	135	*Posting thoughts in text form* (-) *@CentralFics is a website's official account. The other four are personal accounts of individuals.* (-)	*proud directioner fun weird crazy loud chick ..... follow back 100 hons fine art commonly known deedee lover thor criminal minds minions way life secret member expendables*
20	1625	*NONE* (-) *1, 2, 4, and 5 all seem to be sports fans/tweet about sports. 3 doesn't.* (+)	*union local columbus .... lifelong buckeye fan ... usher ... pete rose belongs hof ironman finisher crossfitter veteran father kcco love dallas cowboys till short live fun !!!*
21	56	*The other four seem a lot more mature based on their posts.* (+) *They are all white people.* (+) *@SarahYogiKAY is the only non-individual account. All other accounts are personal accounts while this belongs to an yoga class / institute* (-)	*one follow follow back cares #teamaquarius #teamidgaf !!!! every tweet cry help !!! kay yoga brings songs games creative stories together teach kids ages joys yoga fun way learn mindful centered*
22	368	*They're all individuals* (+) *Young people* (+)	*collins winter springs professionals strive reduce stress worry associated going dentist*
23	243	*he's an athlete* (+) *all are personal accounts* (-)	*fanbase indonesia share hottest news i'm gentle man caring loving importantly i'm god's fear pour another one pianist singer artist luhan*
24	4680	*The others seem more happy with their lives* (-) *The others are all women.* (-)	*nobody back start new beginning anyone start today make new ending !!!! thoughts create reality keep positive stay lifted sinner who's probably gonna sin lord forgive insta crew — bands — rock roll — carson*

25	1568	*734477232 / @CornishAccounts is the only business account of a chartered accountant firm. The rest of the four are personal accounts of individuals.* (+) *The other 4 are people not businesses* (+)	*chartered accountants business advisors gold xero partners cornwall xero accounting partner year 2015 south tweets don't see things see wife mommy three handsome boys seeking following god wonderful blessed life written stars junkie mad men addict @howardu reinventing status quo htown beyond*
26	114	*The first one is a website's twitter, the other 4 are personal accounts.* (-) *NONE* (+)	*dance list ceo entrepreneur technology evangelist sedang dan menerima sesuatu istiqomah faculty nursing sky green*
27	454	*The rest are not commercial accounts.* (-) *none* (-)	*texan tarleton state crossfitter biology major justen casual youtube gamer looking bring comedy fun within call duty naruto game play love video games anime wwe mlp reading manga holiday rental video tours web video marketing services rental owners agents extend market reach increase bookings love hanging friends going amusement parks obsessed coasters love good time*
28	319	*The others are female Indonesians.* (+) *Keeping track of followers via bots* (+)	*vida april 1993 sus ojos eran mova con saben muchas cosas xliii spn pwt jangan*
29	1785	*The other 4 are people not businesses* (+) *Talking about a variety of issues but not focused on followers* (-)	*account berbagi informasi seputar teknik komputer politeknik negeri jakarta care share #team dreambig los das lluvia agua las calles cabeza*
30	1721	*Female* (+) *NONE* (+)	*striving towards best self laugh day you'll live happy life god things abigail sierra potter*

31	1303	*none* (-) *Speak English* (-)	*tanner things possible bittersweet like caramel he's broken cause ready everything*
32	3420	*The other profiles are all in foreign languages.* (-) *Foreigners tweeting in their native languages, but not bots* (-)	*panggil aja yan yang juga boleh anthony santos estudiante del #19 mundo mil*
33	7094	*He has a much greater amount of followers, which is 1,282. Also, he is following so many more people, 1,700* (-) *The others seem to be males* (+)	*rings like bell night wouldn't love love instagram technology radio geek content controller northsound bauer media views professional pool spa care csp afo cpo certifications dad firefighter blogger rocker new block party !!!*
34	510	*The other four seem like real people, not bots* (-) *Communicating with a world community, not just close friends* (-) *They aren't afraid to show pictures of themselves in their profile pics.* (-)	*disfruta cada dia como ultimo regular dude give take world got takes fsu*
35	865	*other four are young women* (-) *The other four are real people* (+) *no human photo* (-)	*berks summer 2013 real food recipes natural living side sass ese momento ella beso mundo one direction justin bieber demi lovato jonas brothers cd9 5sos miley cyrus*
36	3627	*A group tweeting lots of business and/or promotional information* (-) *the others speak the english language* (+) *The other four involve people that speak english* (+)	*londoners dad hobbit photographer project manager android enthusiast round gadget lover account berbagi informasi seputar teknik komputer politeknik negeri jakarta care share nyla marie #proudmom*

37	191	*Posting daily horoscope links* (+) *other four are women* (-)	*lanta ain't got bait takes hook #teamvirgo #teamsexy #teamnosleep welcome lair ... it's filled cosplay cars princesses animals food sexy magic enjoy*
38	8309	*none* (-) *Young people* (+)	*sk8 don't trip mind business stay lane bahamas 242 r.i.p law graduate aspiring solicitor*
39	2318	*Young people* (+) *none* (-)	*keperawatan nurse mahasiswa tingkat akhir ambitious work hard play harder partner construction inc snap acres*
40	1139	*the others are male* (+) *A group of people tweeting about politics and general interests* (-) *The other profiles seem to be involved in sports somehow (soccer)* (+)	*rings like bell night wouldn't love love instagram sports physiologist love liverpool sports general #jft96 #ynwa #afc @arsenal*
41	681	*A group of people tweeting personal thoughts and hobbies* (-)	*hit white ball around field sometimes good days i'm charming fuck multifandom designer individualist infantryman*
42	926	*This one is not a full profile* (-) *All seem to have an interest in popular music.* (-) *the rest are in english* (-)	*una lugar donde usted puede perder sin perder kind happened life short unhappy smile darn*
43	1589	*This user is in a foreign language, maybe Spanish and the others are in English.* (-) *The others did ot say they like britney spears* (+) *Personal accounts talking about their interests and lives* (-)	*tengo tantas cosas que decir que ahogo parece ser que dolor como tesoro god sapiosexual ... love music frank ocean yamo pisces sharing love local 1st gift card exclusively independent shops services near amante vida hacer rer gente ser buen hijo hermano amigo gran fan @britneyspears*

44	2446	*All are young women, college age or somewhat older.* (-) *the others seem like they are not very adventurous* (-)	*nyc positive fab*
45	1786	*The other 4 are not black.* (-) *The other four are young* (+) *People sharing personal non-celeb related thoughts and quotes* (-)	*follower christ texan paige baby| finnish girl loves beauty stuff good music ice hockey ..... never said perfect nobody walkin earth ....*
46	1673	*They appear to be nothing more than bots* (+) *NONE* (-)	*without struggle progress handsome clever boy always liked boy true humorist real hustler ontop game gang smart gorgeous gurl gurl ain't gat tym type bio buh willing knw knw wah ryt #team gemini #peace*
47	650	*Foreign accounts talking about life in their native language* (-) *different language* (-) *They all have English as common but the user 5 in some other language.* (-)	*instagram hear birds summer breeze inna maal yusra optimis sesuatu yang tapi tetap*
48	702	*User 4 is a man, the rest are women. Also, user 4 is the only one who mentions that he is a parent.* (-) *NONE* (-)	*actions good thoughts positive energy speak louder judgmental words powerful tools use working toward better world felicidad esta dentro uno ado nadie texas beer gypsy time's person year 2006 opinions sales marketing manager red caboose winery*
49	593	*business* (+) *Personal accounts sharing opinions, mundane details, and interests* (+)	*arte hey everyone god joy strength life stay faithful*

50	689	*They have in common that they can understand foreign language and there is no English in their comments under the profile. They might not be native Americans.* (+) *Foreign accounts tweeting in their native languages* (+)	*fotgrafo reportero del centro argentino estudiante perro bad life bad day desde 2011 hasta ely sueo importa quien quiera antes querido gira tanto nyc life home sweet home glad back*

Table A.1: **PCA on ego text** – embedding cluster labels.

PCA on All Views

Cluster Index	Number Members	Labels	User Text
1	847	*They are all young people* (-) *none* (-)	*aos del lectora tributo lll amor que mereces ser living dream osu student*
2	3563	*The others are more about american culture* (-)	*boyes*
3	925	*The others are women* (-) *Young women tweeting about female interests* (-)	*sabe como sempre fui uma pessoa ... health travel enthusiast cisco zeal thoughts donate team savannah*
4	1075	*none* (-) *They are all young, hip people* (+)	*#teamfollowback .... tweet 5'0 earth chick named would tweet die bury inside gucci store live life like theres tomorrow regrets love everything cause never know last moment*
5	710	*@_AskMeIf_IGAF appears to be a bot as most tweets are automated. The other four are posting own tweets* (-) *The other four are less offensive* (-) *Communicates interest in pop culture and personal tidbits* (-)	*snapchat siempre todo ingeniero civil obras muy esto catalan corazon jaja #teamvirgo #teamfollowback ... catch .!! cant love hoe make weak ... 7teen cal family friends anything sarcastic*
6	1937	*The other accounts have more substance* (-) *Tweets from follower counters and picture platforms* (-)	*desde los con una locura una felicidad mundo redondo pin dreamer hopeless romantic poet ass*
7	3069	*The other 4 are adults who speak english* (+) *Group sharing personal tidbits, unprofessionally* (-)	*southern seaside newcastle supporting teacher ict geography leader like chocolate travelling architect rise ummah dalton bean*
8	114	*none* (-) *Identical bots* (+)	*everything i'm made everything*

9	4435	*none* (-) *Foreign people tweeting in their native languages* (+)	*never don't mind thing brooklyn based street artist charity worker owner little soapy secrets empire mother wife love life love work hard play harder amo mucho*
10	1982	*The other four seem like actual people* (-) *These users type in English.* (-) *Men talking about their lives and interests* (-)	*husband father friend student artist soon game design porno star prev life give dey want make feel like dey need #teamleo bout one character middle finger fanbase @mi_christychibi keep calm support @cherrybelleindo 4ever admin bff*
11	35	*Women tweeting about personal issues and interests* (-) *They are all girls* (-)	*dire che come dio non dadi non credo nelle theres certain happiness bein silly rediculous time succes comes work dictionary*
12	126	*Scorpios interested in astrology* (+) *The others did not post about soccer.* (-)	*trust select i'm filipina bitch experience*
13	3029	*@PauliiiTCA is the only one tweeting in a non-english language (spanish). The other four users are tweeting in english.* (-) *the rest appear to be male* (-)	*ian mitchell phd hcpc swansea city afc wales national team performance psychologist hakuna matata i'm singer/songwriter unsigned artist also follow instagram sexo son taylor lautner @avrillavigne @coldplay @eminem*
14	7191	*They are all women* (-) *none* (+)	*#teamfollowback surgical tech student aspiring model/actress ultimate dreamer naked dude squash libra*
15	431	*Personal accounts not tweeting bot stats* (+) *The other accounts feel more authentic* (+)	*artist specializing drawing sculpture printing graphic design ironic memer youtube star never give fix mistakes keep stepping star ambassador*

16	4615	*might be a business* (-) *Males discussing their lives and opinions* (-)	*freedom isn't free getting money francophone instagram live life like boss wedding prewedding service*
17	102	*Girls talking about music and makeup* (-)	*auntie future superstar forever want maccies vivir con fuerza locura libertad msica alma instagram follow instagram put mind achieve anything*
18	150	*The one that doesn't belong is a family with kids and the others are young and probably single.* (+) *the others are adults, these are children* (-) *Young adults expressing their opinions and interestes* (+)	*motivation christ follower husband father friend bonito encontrar amor vida todos los das misma persona quiero sonrer siempre lado kidrauhl*
19	1103	*The other profiles did not talk about band or sports* (-) *@DenunaArifandi is the only user tweeting in non-english language. He is an Indonesian.* (-)	*free thinker wine drinker par t-t ime investor occasional putt sinker views retweet necessarily endorsement follower christ husband beautiful wife daddy amazing kids student discipleship pastor coffee lover stl cards indy colts fan love guns roses wwe big cubs fan huge fantasy football baseball player loves laugh trys things*
20	6867	*the others are young* (-) *The others are coaches or motivational people.* (-)	*worrying wont stop bad happenin stop good bein enjoyed ... mac pro athlete #broncos #heatnation jerz e-book blogger information take areas life average extraordinary since 1912 helping people organizations achieve outstanding results professional daily lives sr. media strengthening specialist #journalist #trainer*

21	56	*He is a male who doesn't use English. The others are English-speaking females.* (-) *Young women tweeting about their lives* (-) *The others show off less skin* (-)	*trinity middle storm allstars #directioner #lovatic #1dfamily #mixer everything beautiful time rip lcpl nick forever hearts cupcakes baker music lover dancing queen pink princess marketing major runner wellness blogger lover laughter sports lover indian girl*
22	1057	*They are american.* (-) *Communicating sports or pop culture interests* (-)	*goin put gay saying taylor love pink frank ocean cheetah print jake family friends 412 tamo*
23	560	*The other four seem more easy going and laid back* (-) *The others seem well adjusted* (-) *A group of foreigners tweeting in different languages* (-)	*born raised dot say what's mind alot times that's random shit drag make always looking new opportunity adventure say bachelor engineering dude want prove words*
24	368	*2-5 are all people, while number one seems to be some kind of website.* (-) *none* (-)	*vivir con fuerza locura libertad msica alma instagram corse telecom cinma entrepreneur ces tweets que moi*
25	1195	*Speak English* (+) *Women, possibly all Hispanic* (-)	*menos como forma vida lado los realidad donde sea pero con verdadero hermanas cosplayer full-time art history student otl daily astrology vivian owen author lucky stars astrology bringing ancient star wisdom modern-day life featured writer*
26	1348	*Communicating information relevant to their person or business instead of bots* (-) *The others re all individuals* (-)	*husband literature enthusiast sga bee gees dadakan baca horison sekali lagi seneng lari don't live something you'll die nothing snap hospital gato projeto que felina horas por dia life short unhappy smile darn*

27	80	*They appear to be mostly bot posts* (+) *none* (+)	*arsenal life hallo proud directioner fun weird crazy loud chick ..... follow back 100 better god i'm promise xii path putri*
28	684	*This person seems famous and the others don't* (-) *The others seem like young people who haven't figured out who they want to be in life yet.* (-)	*saya ank dari competition striving better woman yesterday amikom lillywhite temen work make giving*
29	3733	*Men tweeting about various issues besides music* (-) *This person is very vague and the others describe themselves better* (+)	*... books yarn that's life rip shanell gone never forgotten love 4ever brothers otf ces dames pote tel papillon*
30	1801	*Interested in pop culture and celebrities* (-) *The other 4 seem more conservative and relaxed* (+)	*huge mma fan undercover ninja phat suit improving skillz daily day ninja appear without suit avenge belieber forever belieber love justin drew bieber !!! download @shots tell friend we're alive free conoces orgullosa justin drew bieber mallette demetria devonne lovato hart enamor con*
31	665	*Looks like a fan account* (+) *the others are women* (-)	*poeta palabras tan del estudio salud vida ... fanbase @mi_christychibi keep calm support @cherrybelleindo 4ever admin bff mirror lie shows what's inside*
32	86	*Personal accounts of guys tweeting about their lives* (+)	*hustle like starving going hard gotta eat smiling happy gay guy trying humble best possible way smile face times twitter tumblr blog fangirls guide everything hamdan bin mohammed bin rashid maktoum don't fucks many catch solo dolo #teamcapricorn*

33	2528	*This one is male and a foreign musician. The others are females and pretty average.* (-) *the rest are women* (-)	*fearfully wonderfully made psalm 139:14 fitness coach blessed beyond belief diggin deeper everyday canadian raised i'm competition — dream chaser getter idk idc bastille 1975 bangladeshi musician*
34	975	*A group of college girls from around the world* (-) *The other four are women* (-)	*music life god geronimo anderson jr. typical cute chick international relations fisip universitas katolik bandung manajemen usu impossible line aditya pratama*
35	5832	*Young women talking about their lives and female issues* (-) *They are all female.* (-)	*founder club live laugh love bartender workaholic random follow follow back twitter 0/4 0/5*
36	81	*none* (-) *Male* (+)	*kay yoga brings songs games creative stories together teach kids ages joys yoga fun way learn mindful centered par t-t ime home brewer full time beer geek follower christ husband tech nerd awesomesauce occupied things designed brewed necessarily often overlapping hub universe*
37	704	*The other four seem much skinnier* (+) *This one is the only one in English* (+) *This guy is the only English speaking person.* (+)	*alvin sederhana tapi disfruta momento springsteen fanatic pittsburgher penn stater wwe fan*
38	1095	*A foreign group of people tweeting in their native languages* (-) *the others are women* (+) *these four users are you* (+)	*allah art pharmacist born sleep stalker*

39	1265	*NONE* (-) *People likely looking for a job* (-)	*estudiante comunicacin audiovisual #happiness tdcs shortage fault found amid stars ankara university school medicine things important make it's mind mind control ig-*
40	635	*The other users aren't smoking in their profile pic.* (+) *Personal anecdotes and information* (-) *all others are English* (-)	*getting money nwmsu omaha things important make it's mind mind control ig- careful hurt could ruin life*
41	1667	*Personal accounts talking about their interests and opinions* (+) *based on the images and what I can see he is not from not caucasian* (+) *The other accounts are much older* (-)	*madrid #ravensnation yamaha banshee oregon dunes*
42	154	*Foreigners in native languages talking to a wide audience* (+) *The other four seem to live a more American-like culture* (-)	*sometimes fun bad things ponta portugal realmadrid allah always sma negeri enjoy moment still last comedian lovers*
43	9230	*They all have over 100 followers.* (-) *this person seems like a comedy account while the others seem real* (-)	*texan tarleton state crossfitter biology major justen always thinking one meal ahead mig man bor uppsala ._. snapchat lite shr och nerd viking engineering student destroyer worlds*
44	9907	*Girls talking about female issues and their lives* (-) *the rest aren't from Australia* (+)	*stop worrying take chance heyy year old aussie life taking life comes trying let anything get high school yeah*

45	3685	*The other accounts are actual people* (-) *Personal accounts tweeting issues personal to them* (-)	*integrated medicine center aiming holistic improvement health bring efficient alternative complementary treatments city ward citizen helpers inc elder care compete remodeling services advisor*
46	2568	*Tweeting mostly in text, not just pictures from another platform* (-) *this looks like the only fan account* (-)	*fan account dez duron spreading word talented artist tea enthusiast dreams black white rnb pizza lover| dancer singer*
47	476	*An interest in religion, particularly, Jesus* (+) *Christians making mention of their faith* (+)	*may boast christ crucified alone son student sunday school teacher bible study leader remain love john lover jesus president walk incorporation*
48	1195	*Men tweeting about non-professional lives and interests* (-) *The other four seem older* (-)	*came drink milk kick ass i've finished milk one lab accident away super villian basketball writer china press man loves basketball #lfc #ynwa #faith #believe #basketballneverstops official fanbase account fitri*
49	5880	*Largely anonymous and impersonal, tweeting about specific interests* (-)	*belieber like mauve*
50	512	*none* (-)	*#blessed fans final fantasy kingdom classic*

Table A.2: **PCA on all views** – embedding cluster labels.

dGCCA on All Views

Cluster Index	Number Members	Labels	User Text
1	48	*the others are non business types* (-) *A group of male sports fans* (-)	*huge cleveland sports fan i'm absolutely hilarious least made hawaii giants 49ers snap chat instagram gomez class 2012 world cup 2014 dream come true #mexico god family saint mary's alumnus poker junky state farm insurance*
3	57	*the others seem to be more young* (-) *Personal opinions on life and culture* (+)	*kinda shy unpopular kid lover music i'm going broadway knw i'll bak #weare psu panda lover music lover fall boy hell yah !!!! maroon5 take concert things christ strengthens -philippians 4:13*
4	80	*the other four are skinnier women* (-) *Young women talking about love and partying* (-)	*hunters foxes live live right enough life short regrets happily married mummy got perfect husband feel blessed crazy little life*
5	32	*this one appears to be a musician* (-) *Foreign accounts tweeting in their native language* (+) *the others are female* (-)	*fight achieve dreams fearfully wonderfully made psalm 139:14 fitness coach blessed beyond belief diggin deeper everyday rest peace boo baby sister always love rhymes make sick*
6	594	*The others are individual people* (-) *They are in one form or another involved with Christianity or a church.* (+) *Christians expressing their faith* (+)	*generations christan church connecting people christ christ follower husband father friend nyc queen's palace chosen enter i'm royal priesthood conqueror sick broke blessed friend jesus*

7	112	*Female-centric tweets and interests* (-) *The others have more of a human touch* (-)	*non-profit providing services job seekers employers #followback updates career events advice openings job fairs currently starring reality show titled modern cinderella one girls search love shoes super mom business owner pageant girl host cool chick radio take inches single treatment instant body sculpting results natural without surgery*
9	182	*The others are young* (-) *The other four are younger individuals in their 20s* (-)	*#teamshady #teamfollowback #stan live music moments nobody lives forever life really really short instagram female rebel native pride call orange grove high school senior*
10	92889	*Girls directly tweeting thoughts relevant to their personal lives* (+) *The other accounts speak english* (-)	*gonna eat tots beware suss*
12	272	*the other accounts dont seem to represent individual real people.* (-) *There's one other business account, but for the most part the other four are personal and political while this one is promotional* (-)	*alberta top notch movers moving needs edmonton please contact website call work tirelessly provide clients high quality language branding solutions language problem follow back engineer/ scholar love #pakistan living kpk*
13	322	*Number 2 is japanese and only tweets in japanese. The others are tweet in English.* (+) *none* (+)	*screaming guitar dark humor jack jameson recording engineer training hard shell soft heart never regret mistake made one point it's exactly wanted ... bbm pin thought toaster possessed put poptart caught fire*

14	62	*These users type in English.* (-) *Personal accounts talking about their lives in a personal way* (-)	*#warrior someday .... #belieber #mahomie #begreat currently starring reality show titled modern cinderella one girls search love shoes mejor sin duda fan del mejor dolo del mundo*
15	148	*The others seem like mature adults* (+) *They're all women with very positive attitudes.* (-) *A group of younger people tweeting mundane details about personal matters* (+)	*disneyland cast member attractions enthusiast english studies student stirling uni florida snapchat instagram never get wet never expose bright light never ever feed midnight*
16	130	*Young people posting about their lives* (-) *NONE* (+)	*official account pearl block report spam mentions follow genesis #sagittarius florida snapchat instagram*
17	34	*This user is male and all the others are female* (+)	*rings like bell night wouldn't love love instagram nigeria base rapper song writer composer cash flow entertainment ... booking contact three*
18	376	*The others are women* (-) *A group of girls tweeting random thoughts* (-) *last four seem to be women* (-)	*love even though famous still humble boy canada proud #belieber saw july 10th 2013 llamo cami encanta justin perfecto justin years basketball hiphop jazz i'm awesome*
19	136	*none* (-) *The four users are all young males and seem to have stereotypical "male-related" interests like sports, gaming, etc.* (-)	*follow new account football cooking sleeping life ... yeah playing xbox miss understanding follow i'll follow back :))) love guns roses wwe big cubs fan huge fantasy football baseball player*

20	736	*NONE* (-) *They are all young people* (-)	*auntie future superstar forever longtime fitness enthusiast big time lover certified personal trainer holistic nutritionist wannabe runner yogi self taught tea guru photograper dreamer fashion media student fashion stylist argentina alegra sueo gracias @donniewahlberg*
21	37	*the four other seem to be younger women, this one has a panda and is in korean.* (-) *Girls talking about their lives and interests* (-) *different language than the others* (-)	*mostly tweet pancakes disney like people let's friends carpe diem ball light reborn christ shine even darkest nights rowan instagram snapchat*
22	96	*@AtikahLey appears to be bot as all her tweets are automated. The other four post their own tweets* (+) *the others are women* (-) *A group of people with bots to check their twitter stats and followers* (-)	*allah mbf instagram love guys cooking nascar dale #88*
23	121	*none* (-) *Gay men* (-)	*amante msica gym lady gaga little monster born way valley boy life play puck since giver get fuck way follow beauties*
24	21	*Female* (-) *NONE* (+)	*don't like someone try walking mile shoes still don't like you're mile away shoes jackson followed wesley stromberg followed kenneth lowery*

25	228	*business* (-) *Socially conscious and inspiring accounts* (-)	*amo mis amigos familia buena msica feminista atea poco loca estudiante hincha racing certified teach dr. tai chi health programs teacher various locales write column home page blog meditation motion stay young wild one hunters foxes*
26	55	*Seems to be the only English speaker.* (+) *They are all foreign speaking individuals who live somewhere in the middle East.* (+) *Foreign women tweeting in their native language* (+)	*gonna eat tots 10% genius 90% idiot everlasting @siwon407 endless @allrisesilver 94l admiring without desiring empty .....*
27	62	*A group of English speaking sports and motivational quote fans* (+) *He seems older than the others* (-)	*would attempt knew could fail |19| you're favorite muggle fangirl band members youtubers*
28	476	*A group of females with inspirational/ health related quote and interests* (+) *language* (-)	*oos letras profesional sarcasmo mierda living life man either like don't care life*
29	59	*She doesnt look like shes smiling in her profile pictures like the others. The other 4 have better bios. She doesnt tweet in English at all while the others do* (-) *language* (-)	*line path nida tanjung striving towards best self committed constituents life percussion*
30	69	*none* (-) *Real accounts, not fraud/bots* (-)	*lively disposition delighted anything ridiculous beautiful life short live full time #vip #bbc positive mind positive vibes positive life pain discipline better pain regret*

31	2502	*A diverse group of people tweeting about their diverse lives, not just one specific interest* (+) *The other 4 are young* (-)	*conservative minded free thinker loves god guns country retired usaf snapchat kik belum baik dan pengen jadi baik kalau pun udah baik pengen dan harus lebih baik gaeilge teo tones*
32	286	*Sharing personal thoughts, not follower information* (-) *rest are women* (-) *The other four are young women* (-)	*instragram .... vuelvo dormir ... soy tipo simple ser feliz con locura mujer sentimental*
34	57	*Young women talking about their lives and views on the world* (+) *most creative picture* (+)	*god family dacc nursing major dreams come true courage pursue instagram xxiv*
35	91	*only company* (-) *The other four feel like real people and this account feels like its meant to push promotions* (-) *People talking about pop culture and entertainment as opposed to personal lives* (+)	*breathe live carry follow*
37	205	*NONE* (-) *Men* (+)	*yid end come #coys #ukip #ttid #coys smart gorgeous gurl gurl ain't gat tym type bio buh willing knw knw wah ryt #team gemini #peace i'm matty space cowboy billionaire liar ... instagram snapchat #thfc #coys*
38	36	*The others are non business accounts* (+) *Personal accounts talking about their lives and views* (+)	*exceptional representation defending rights children interests times 877 worldwide dirtbag #23 #bayarea #dubnation lively disposition delighted anything ridiculous*

39	176	*All seem to live in the United States of America.* (-) *the others seem more happy with life* (-) *The others are females with pretty basic female interests unlike the male with a political bent.* (-)	*changes wish see world mahatma ghandi bad 90s boy bands diet coke time succes comes work dictionary tired called relationship*
40	382	*the others have a profile with a human touch* (+)	*capture dream make real 5sos fall boy screamau directioner july 2010 niall harry louis graduated high school currently attending year college love gaming enjoy making videos interacting subs that's*
41	48	*none* (+) *The 4 profiles belong to girls, number 3 is a guy.* (+)	*leaders always followers it's guy going walk here's good times keep smiling change someones day*
43	141	*The other four seem to be more conservative* (-) *the other four are regular people, this appears to be a musician* (-) *Foreign accounts tweeting in their native languages* (+)	*wife mom esthetician chocolate lover book reader embarrassing dancer slightly ocd someday traveler life lover always moments notice away crazy mature honeybee art way run away without leaving home ask.fm:*
44	41	*The others are more into american culture* (-) *Young men tweeting about personal lives and interests* (-) *The other 4 use one language while this account uses at least two* (-)	*use smile change world don't let world change smile panam #yrn invisible family*
45	43	*A group of foreign people tweeting text and words, not just pictures* (-) *The other 4 seem to be more dedicated to Hispanic culture* (+)	*avila tierra que sabe cantante guitarrista jornada completa con arriba tengo corazn que llora por que disfruta cada dia como ultimo kismet roll dice raise stakes leave rest faith*

46	223	*none* (+) *1, 3, 4, and 5 are girls. 2 is a guy.* (+)	*mexicano aos grammar nazi enamorado del oblivion las mujeres que idk idc bastille 1975 fab*
47	38	*The others seem like more independent individuals.* (-)	*cant say something nice dont say nothing student brother son athlete thinker i'm awesome*
48	177	*Food related businesses* (+) *They are all twitter accounts for adults or businesses.* (+) *All promoting businesses.* (+)	*proud winners best product year 2013 food producer year 2012 hampshire life food drink awards we're traditional microbrewery based heart kent tenterden taekwondo dream try pray success consultant publisher writer rock roll singer norwich ambassador norwich tourist guide*
49	47	*All of them have profile pictures of themself exvept the one picked* (-)	*living dream snapchat*
50	53	*none* (-) *English speakers* (-)	*add snapchat instagram kik telegram mad cat lady i'm half walrus half potato walk new way hip hop*

Table A.3: ***dGCCA* on all views** – embedding cluster labels.

Bibliography

Abel, Fabian, Qi Gao, Geert-Jan Houben, and Ke Tao (2011). "Analyzing User Modeling on Twitter for Personalized News Recommendations". In: *User Modeling, Adaption and Personalization - 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings*, pp. 1–12. DOI: 10.1007/978-3-642-22362-4_1. URL: http://dx.doi.org/10.1007/978-3-642-22362-4_1.

Adomavicius, Gediminas and Alexander Tuzhilin (2005). "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions". In: *IEEE Transactions on Knowledge & Data Engineering* 6, pp. 734–749.

Ahmed, Amr and Eric P Xing (2010). "Staying informed: supervised and semi-supervised multi-view topical analysis of ideological perspective". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1140–1150.

Al Zamal, Faiyaz, Wendy Liu, and Derek Ruths (2012). "Homophily and Latent Attribute Inference: Inferring Latent Attributes of Twitter Users from Neighbors." In: *ICWSM* 270, p. 2012.

Amir, Silvio, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva (2016). "Modelling context with user embeddings for sarcasm detection in social media". In: *arXiv preprint arXiv:1607.00976*.

Anand, Pranav, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor (2011). "Cats rule and dogs drool!: Classifying stance in online debate". In: *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*. Association for Computational Linguistics, pp. 1–9.

Andrew, Galen, Raman Arora, Jeff Bilmes, and Karen Livescu (2013). "Deep canonical correlation analysis". In: *International Conference on Machine Learning (ICML)*, pp. 1247–1255.

Aral, Sinan, Lev Muchnik, and Arun Sundararajan (2009). "Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks". In: *Proceedings of the National Academy of Sciences* 106.51, pp. 21544–21549.

Aramaki, Eiji, Sachiko Maskawa, and Mizuki Morita (2011). "Twitter catches the flu: detecting influenza epidemics using Twitter". In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 1568–1576.

Arora, Raman and Karen Livescu (2014). "Multi-view learning with supervision for transformed bottleneck features". In: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, pp. 2499–2503.

Arora, Raman, Teodor Vanislavov Marinov, Poorya Mianjy, and Nati Srebro (2017). "Stochastic Approximation for Canonical Correlation Analysis". In: *Advances in Neural Information Processing Systems*, pp. 4778–4787.

Ayers, John W, Theodore L Caputi, Camille Nebeker, and Mark Dredze (2018). "Don't quote me: reverse identification of research participants in social media studies". In: *Nature Digital Medicine* 1.1, p. 30.

Bach, Francis R and Michael I Jordan (2005). "A probabilistic interpretation of canonical correlation analysis". In:

Bakker, Bart and Tom Heskes (2003). "Task clustering and gating for bayesian multitask learning". In: *Journal of Machine Learning Research* 4.May, pp. 83–99.

Bakshy, Eytan, Solomon Messing, and Lada A Adamic (2015). "Exposure to ideologically diverse news and opinion on Facebook". In: *Science* 348.6239, pp. 1130–1132.

Bakshy, Eytan, Jake M Hofman, Winter A Mason, and Duncan J Watts (2011). "Everyone's an influencer: quantifying influence on twitter". In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, pp. 65–74.

Beller, Charley, Rebecca Knowles, Craig Harman, Shane Bergsma, Margaret Mitchell, and Benjamin Van Durme (2014). "I'm a belieber: Social roles via self-identification and conceptual attributes". In: *ACL*. Vol. 2, pp. 181–186.

Benton, Adrian, Raman Arora, and Mark Dredze (2016). "Learning multiview embeddings of twitter users". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Vol. 2, pp. 14–19.

Benton, Adrian, Glen Coppersmith, and Mark Dredze (2017). "Ethical research protocols for social media health research". In: *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pp. 94–102.

Benton, Adrian and Mark Dredze (2018a). "Deep Dirichlet multinomial regression". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Vol. 1, pp. 365–374.

— (2018b). "Using Author Embeddings to Improve Tweet Stance Classification". In: *Proceedings of the Workshop on Noisy User-generated Text*.

Benton, Adrian, Margaret Mitchell, and Dirk Hovy (2017). "Multitask learning for mental health conditions with limited social media data". In: *15th EACL*. Vol. 1, pp. 152–162.

Benton, Adrian, Braden Hancock, Glen Coppersmith, John W Ayers, and Mark Dredze (2016a). "After Sandy Hook Elementary: A year in the gun control debate on Twitter". In: *arXiv preprint arXiv:1610.02060*.

Benton, Adrian, Michael J Paul, Braden Hancock, and Mark Dredze (2016b). "Collective Supervision of Topic Models for Predicting Surveys with Social Media". In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2892–2898.

Benton, Adrian, Huda Khayrallah, Biman Gujral, Dee Ann Reisinger, Sheng Zhang, and Raman Arora (2017). "Deep generalized canonical correlation analysis". In: *arXiv preprint arXiv:1702.02519*.

Bermingham, Adam and Alan Smeaton (2011). "On using Twitter to monitor political sentiment and predict election results". In: *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pp. 2–10.

Bingel, Joachim and Anders Søgaard (2017). "Identifying beneficial task relations for multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1702.08303*.

Bjerva, Johannes (2017). "One Model to Rule them all: Multitask and Multilingual Modelling for Lexical Analysis". PhD thesis. University of Groningen.

Blei, D., A. Ng, and M. Jordan (2003). "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research (JMLR)*.

Blitzer, John, Mark Dredze, and Fernando Pereira (2007). "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification". In: *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447.

Blum, Avrim and Tom Mitchell (1998). "Combining Labeled and Unlabeled Data with Co-Training". In: *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100.

Bollen, Johan, Huina Mao, and Xiaojun Zeng (2011). "Twitter mood predicts the stock market". In: *Journal of computational science* 2.1, pp. 1–8.

Boslaugh, Sarah E, Matthew W Kreuter, Robert A Nicholson, and Kimberly Naleid (2004). “Comparing demographic, health status and psychosocial strategies of audience segmentation to promote physical activity”. In: *Health Education Research* 20.4, pp. 430–438.

Bottou, Léon (2012). “Stochastic gradient tricks”. In: *Neural Networks, Tricks of the Trade, Reloaded*, pp. 430–445.

Bromley, Jane, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah (1994). “Signature verification using a" siamese" time delay neural network”. In: *Advances in neural information processing systems*, pp. 737–744.

Brown, Peter F, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai (1992). “Class-based n-gram models of natural language”. In: *Computational linguistics* 18.4, pp. 467–479.

Cadwalladr, Carol and E Graham-Harrison (2018). “The Cambridge Analytica Files”. In: *The Guardian. Retrieved Mar* 17, p. 2018.

Callison-Burch, Chris and Mark Dredze (2010). “Creating Speech and Language Data With Amazon’s Mechanical Turk”. In: *NAACL-HLT Workshop on Creating Speech and Language Data With Mechanical Turk*, pp. 1–12.

Card, Dallas, Chenhao Tan, and Noah A Smith (2018). “Neural Models for Documents with Metadata”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 2031–2040.

Carroll, J Douglas (1968). “Generalization of canonical correlation analysis to three or more sets of variables”. In: *Convention of the American Psychological Association*. Vol. 3, pp. 227–228.

Caruana, Rich (1996). “Algorithms and applications for multitask learning”. In: *ICML*, pp. 87–95.

— (1997). “Multitask learning”. In: *Machine learning* 28.1, pp. 41–75.

— (1993). “Multitask Learning: A Knowledge-Based Source of Inductive Bias”. In: *Proceedings of the Tenth International Conference on Machine Learning*, pp. 41–48.

Caruana, Rich, Shumeet Baluja, Tom Mitchell, et al. (1996). “Using the future to “sort out" the present: Rankprop and multitask learning for medical risk evaluation”. In: *Advances in neural information processing systems*, pp. 959–965.

Chandar, Sarath, Mitesh M. Khapra, Hugo Larochelle, and Balaraman Ravindran (2015). “Correlational Neural Networks”. In: *CoRR* abs/1504.07225. URL: http://arxiv.org/abs/1504.07225.

Chang, J., J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei (2009). “Reading tea leaves: How humans interpret topic models”. In: *NIPS*.

Chen, Kailong, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu (2012). "Collaborative personalized tweet recommendation". In: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 661–670.

Chen, Xin, Yu Wang, Eugene Agichtein, and Fusheng Wang (2015). "A Comparative Study of Demographic Attribute Inference in Twitter". In: *Conference on Weblogs and Social Media (ICWSM)*.

Ciot, Morgane, Morgan Sonderegger, and Derek Ruths (2013). "Gender Inference of Twitter Users in Non-English Contexts". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Wash*, pp. 18–21.

Cohen, Raviv and Derek Ruths (2013). "Classifying political orientation on Twitter: It's not easy!" In: *ICWSM*.

Collier, Nigel and Son Doan (2011). "Syndromic classification of twitter messages". In: *International Conference on Electronic Healthcare*. Springer, pp. 186–195.

Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th International Conference on Machine Learning*. ACM, pp. 160–167.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa (2011). "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12.Aug, pp. 2493–2537.

Conway, Mike (2014). "Ethical issues in using Twitter for public health surveillance and research: developing a taxonomy of ethical concepts from the research literature". In: *Journal of medical Internet research* 16.12.

Coppersmith, Glen, Mark Dredze, Craig Harman, Kristy Hollingshead, and Margaret Mitchell (2015a). "CLPsych 2015 Shared Task: Depression and PTSD on Twitter". In: *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Denver, Colorado: Association for Computational Linguistics, pp. 31–39. DOI: 10.3115/v1/W15-1204. URL: http://aclweb.org/anthology/W15-1204.

Coppersmith, Glen, Kim Ngo, Ryan Leary, and Anthony Wood (2016). "Exploratory Analysis of Social Media Prior to a Suicide Attempt". In: *Proceedings of the Third Workshop on Computational Lingusitics and Clinical Psychology*. San Diego, CA, USA: Association for Computational Linguistics, pp. 106–117. DOI: 10.18653/v1/W16-0311. URL: http://aclweb.org/anthology/W16-0311.

Coppersmith, Glen, Mark Dredze, Craig Harman, and Kristy Hollingshead (2015b). "From ADHD to SAD: Analyzing the language of mental health on Twitter through self-reported diagnoses". In: *Proceedings of the 2nd Workshop on Computational*

Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pp. 1–10.

Coppersmith, Glen, Ryan Leary, Eric Whyne, and Tony Wood (2015c). "Quantifying suicidal ideation via language usage on social media". In: *Joint Statistics Meetings Proceedings, Statistical Computing Section, JSM*.

Crichton, Gamal, Sampo Pyysalo, Billy Chiu, and Anna Korhonen (2017). "A neural network multi-task learning approach to biomedical named entity recognition". In: *BMC bioinformatics* 18.1, p. 368.

Culotta, Aron (2014). "Reducing sampling bias in social media data for county health inference". In: *Joint Statistical Meetings Proceedings*, pp. 1–12.

Culotta, Aron, Nirmal Kumar Ravi, and Jennifer Cutler (2016). "Predicting Twitter user demographics using distant supervision from website traffic data". In: *Journal of Artificial Intelligence Research* 55, pp. 389–408.

Daiber, Joachim and Rob van der Goot (2016). "The Denoised Web Treebank: Evaluating Dependency Parsing under Noisy Input Conditions." In: *LREC*.

De Choudhury, Munmun, Michael Gamon, Scott Counts, and Eric Horvitz (2013). "Predicting Depression via Social Media." In:

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, pp. 391–407.

Dembczynski, Krzysztof, Wojciech Kotlowski, and Dawid Weiss (2008). "Predicting ads clickthrough rate with decision rules". In: *Workshop on targeting and ranking in online advertising*. Vol. 2008.

Ding, Tao, Warren K Bickel, and Shimei Pan (2017). "Multi-view unsupervised user feature embedding for social media-based substance use prediction". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2275–2284.

Dredze, Mark, Michael Paul, Shane Bergsma, and Hieu Tran (2013). "Carmen: A Twitter Geolocation System with Applications to Public Health". In: *AAAI HIAI Workshop*.

Duchi, J., E. Hazan, and Y. Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization". In: *JMLR* 12, pp. 2121–2159.

Eisenstein, Jacob, Noah A. Smith, and Eric P. Xing (2011). "Discovering Sociolinguistic Associations with Structured Sparsity". In: *ACL*.

Evgeniou, Theodoros and Massimiliano Pontil (2004). "Regularized multi-task learning". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 109–117.

Ewerbring, L Magnus et al. (1990). "Canonical correlations and generalized SVD: applications and new algorithms". In: *Advances in Parallel Computing*. Vol. 1. Elsevier, pp. 37–52.

Faruqui, Manaal and Chris Dyer (2014). "Improving Vector Space Word Representations Using Multilingual Correlation". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pp. 462–471. URL: http://aclweb.org/anthology/E/E14/E14-1049.pdf.

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin (2001). "Placing search in context: The concept revisited". In: *Proceedings of the 10th international conference on World Wide Web*. ACM, pp. 406–414.

Fu, Xiao, Kejun Huang, Evangelos E Papalexakis, Hyun-Ah Song, Partha Pratim Talukdar, Nicholas D Sidiropoulos, Christos Faloutsos, and Tom Mitchell (2016). "Efficient and distributed algorithms for large-scale generalized canonical correlations analysis". In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, pp. 871–876.

Fu, Xiao, Kejun Huang, Mingyi Hong, Nicholas D Sidiropoulos, and Anthony Man-Cho So. "Scalable and Flexible Multiview MAX-VAR Canonical Correlation Analysis". In: *IEEE Transactions on Signal Processing* 65.16 (), pp. 4150–4165.

Gammerman, Alexander, Volodya Vovk, and Vladimir Vapnik (1998). "Learning by transduction". In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 148–155.

Gimpel, Kevin, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith (2011). "Part-of-speech tagging for twitter: Annotation, features, and experiments". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pp. 42–47.

Goldberg, Yoav (2015). "A primer on neural network models for natural language processing". In: *arXiv preprint arXiv:1510.00726*.

Goswami, Sumit, Sudeshna Sarkar, and Mayur Rustagi (2009). "Stylometric analysis of bloggers' age and gender". In: *Third International AAAI Conference on Weblogs and Social Media*.

Gu, Quanquan and Jie Zhou (2009). "Learning the shared subspace for multi-task clustering and transductive transfer classification". In: *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. IEEE, pp. 159–168.

Gust, Deborah A., Natalie Darling, Allison Kennedy, and Ben Schwartz (2008). "Parents with doubts about vaccines: which vaccines and reasons why". In: *Pediatrics* 122.4, pp. 718–725.

Guy, Ido, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel (2010). "Social media recommendation based on people and tags". In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 194–201.

Hannon, John, Mike Bennett, and Barry Smyth (2010). "Recommending twitter users to follow using content and collaborative filtering approaches". In: *Proceedings of the fourth ACM conference on Recommender systems*. ACM, pp. 199–206.

Hardoon, David R, Sandor Szedmak, and John Shawe-Taylor (2004). "Canonical correlation analysis: An overview with application to learning methods". In: *Neural computation* 16.12, pp. 2639–2664.

Hardt, Daniel, Dirk Hovy, and Sotiris Lamprinidis (2018). "Predicting News Headline Popularity with Syntactic and Semantic Knowledge Using Multi-Task Learning". In: *EMNLP*.

Harman, Craig, Glen Coppersmith, and Mark Dredze (2014). "Measuring post traumatic stress disorder in Twitter". In: *In ICWSM*.

Hazarika, Devamanyu, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea (2018). "CASCADE: Contextual Sarcasm Detection in Online Discussion Forums". In: *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*.

Hepburn, L., M. Miller, D. Azrael, and D. Hemenway (2007). "The US gun stock: results from the 2004 national firearms survey". In: *Injury Prevention* 13.1, pp. 15–19.

Hill, Felix, Roi Reichart, and Anna Korhonen (2015). "Simlex-999: Evaluating semantic models with (genuine) similarity estimation". In: *Computational Linguistics* 41.4, pp. 665–695.

Hill, Shawndra, Foster Provost, Chris Volinsky, et al. (2006). "Network-based marketing: Identifying likely adopters via consumer networks". In: *Statistical Science* 21.2, pp. 256–276.

Hill, Shawndra, Adrian Benton, Lyle Ungar, Sofus Macskassy, Annie Chung, and John H Holmes (2011). "A cluster-based method for isolating influence on twitter". In: *21st Workshop on Information Technologies and Systems*.

Hollingshead, Kristy (2016). "Detecting Risk and Protective Factors of Mental Health using Social Media Linked with Electronic Health Records". In: *JSALT 2016 Workshop*. Johns Hopkins University. URL: `http://www.clsp.jhu.edu/workshops/16-workshop/detecting-risk-and-protective-`

factors-of-mental-health-using-social-media-linked-with-electronic-health-records/.

Horst, Paul (1961). "Generalized canonical correlations and their applications to experimental data". In: *Journal of Clinical Psychology* 17.4.

Hotelling, Harold (1936). "Relations between two sets of variates". In: *Biometrika* 28.3/4, pp. 321–377.

Hovy, Dirk (2015). "Demographic factors improve classification performance". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Vol. 1, pp. 752–762.

Hovy, Dirk and Anders Søgaard (2015). "Tagging Performance Correlates with Author Age". In: *ACL*.

Hu, Yuening, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith (2014). "Interactive topic modeling". In: *Machine learning* 95.3, pp. 423–469.

Huang, Xiaolei, Xin Li, Tianli Liu, David Chiu, Tingshao Zhu, and Lei Zhang (2015). "Topic Model for Identifying Suicidal Ideation in Chinese Microblog". In: *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*. Shanghai, China, pp. 553–562. URL: http://aclweb.org/anthology/Y15-1064.

Jagarlamudi, Jagadeesh, Hal Daumé III, and Raghavendra Udupa (2012). "Incorporating Lexical Priors into Topic Models". In: *EACL*.

Jansen, Bernard J, Mimi Zhang, Kate Sobel, and Abdur Chowdury (2009). "Twitter power: Tweets as electronic word of mouth". In: *Journal of the American society for information science and technology* 60.11, pp. 2169–2188.

Jiang, Long, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao (2011). "Target-dependent twitter sentiment classification". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pp. 151–160.

Johannsen, Anders, Dirk Hovy, and Anders Søgaard (2015). "Cross-lingual syntactic variation over age and gender". In: *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pp. 103–112.

Kakade, Sham M and Dean P Foster (2007). "Multi-view regression via canonical correlation analysis". In: *International Conference on Computational Learning Theory*. Springer, pp. 82–96.

Kang, Zhuoliang, Kristen Grauman, and Fei Sha (2011). "Learning with Whom to Share in Multi-task Feature Learning." In: *ICML*, pp. 521–528.

Kettenring, Jon R (1971). "Canonical analysis of several sets of variables". In: *Biometrika* 58.3, pp. 433–451.

King, Brian A., Shanta R. Dube, and Michael A. Tynan (2012). "Current Tobacco Use Among Adults in the United States: Findings From the National Adult Tobacco Survey". In: *American Journal of Public Health* 102.11, e93–e100.

Kingma, Diederik and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler (2015). "Skip-Thought Vectors". In: *CoRR* abs/1506.06726. URL: http://arxiv.org/abs/1506.06726.

Konstas, Ioannis, Vassilios Stathopoulos, and Joemon M Jose (2009). "On social networks and collaborative recommendation". In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 195–202.

Kosinski, Michal, David Stillwell, and Thore Graepel (2013). "Private traits and attributes are predictable from digital records of human behavior". In: *Proceedings of the National Academy of Sciences*, p. 201218772.

Kramer, Adam DI, Jamie E Guillory, and Jeffrey T Hancock (2014). "Experimental evidence of massive-scale emotional contagion through social networks". In: *Proceedings of the National Academy of Sciences*, p. 201320040.

Krosnick, Jon A, Charles M Judd, and Bernd Wittenbrink (2005). "The measurement of attitudes". In: *The handbook of attitudes*, pp. 21–76.

Kumar, Abhishek, Piyush Rai, and Hal Daume (2011). "Co-regularized multi-view spectral clustering". In: *Advances in neural information processing systems*, pp. 1413–1421.

Kursuncu, Ugur, Manas Gaur, Usha Lokala, Anurag Illendula, Krishnaprasad Thirunarayan, Raminta Daniulaityte, Amit Sheth, and I Budak Arpinar (2018). "" What's ur type?" Contextualized Classification of User Types in Marijuana-related Communications using Compositional Multiview Embedding". In: *arXiv preprint arXiv:1806.06813*.

Kwok, Irene and Yuzhou Wang (2013). "Locate the Hate: Detecting Tweets against Blacks." In: *AAAI*.

Kywe, Su Mon, Ee-Peng Lim, and Feida Zhu (2012). "A survey of recommender systems in twitter". In: *International Conference on Social Informatics*. Springer, pp. 420–433.

Kywe, Su Mon, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu (2012). "On Recommending Hashtags in Twitter Networks". In: *Social Informatics - 4th International Conference, SocInfo 2012, Lausanne, Switzerland, December 5-7, 2012. Proceedings*, pp. 337–350. DOI: 10.1007/978-3-642-35386-4_25. URL: http://dx.doi.org/10.1007/978-3-642-35386-4_25.

Lai, Pei Ling and Colin Fyfe (2000). "Kernel and nonlinear canonical correlation analysis". In: *International Journal of Neural Systems* 10.05, pp. 365–377.

Lau, Jey Han, David Newman, and Timothy Baldwin (2014). "Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 530–539.

Le, Quoc and Tomas Mikolov (2014). "Distributed representations of sentences and documents". In: *International Conference on Machine Learning*, pp. 1188–1196.

Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg (2010). "Signed networks in social media". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, pp. 1361–1370.

Li, Chang, Yi-Yu Lai, Jennifer Neville, and Dan Goldwasser (2017). "Joint Embedding Models for Textual and Social Analysis". In: *The Workshop on Deep Structured Prediction*.

Li, Jiwei, Alan Ritter, and Dan Jurafsky (2015). "Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks". In: *arXiv preprint arXiv:1510.05198*.

Li, Jiwei, Xinlei Chen, Eduard Hovy, and Dan Jurafsky (2015). "Visualizing and understanding neural models in NLP". In: *arXiv preprint arXiv:1506.01066*.

Liben-Nowell, David and Jon Kleinberg (2007). "The link-prediction problem for social networks". In: *Journal of the American society for information science and technology* 58.7, pp. 1019–1031.

Limsopatham, Nut and Nigel Henry Collier (2016). "Bidirectional LSTM for named entity recognition in Twitter messages". In:

Lipton, Zachary C, David C Kale, Charles Elkan, and Randall Wetzell (2016). "Learning to Diagnose with LSTM Recurrent Neural Networks". In: *Proceedings of ICLR*.

Liu, Wendy and Derek Ruths (2013). "What's in a name? Using first names as features for gender inference in Twitter". In: *Analyzing Microtext: 2013 AAAI Spring Symposium*.

Lohtia, Ritu, Naveen Donthu, and Edmund K Hershberger (2003). "The impact of content and design elements on banner advertising click-through rates". In: *Journal of advertising Research* 43.4, pp. 410–418.

Lu, Chunliang, Wai Lam, and Yingxiao Zhang (2012). "Twitter user modeling and tweets recommendation based on wikipedia concept graph". In: *IJCAI Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*.

Lui, Marco and Timothy Baldwin (2012). "langid. py: An off-the-shelf language identification tool". In: *Association for Computational Linguistics (ACL): system demonstrations*, pp. 25–30.

Luong, Minh-Thang, Hieu Pham, and Christopher D Manning (2015). "Effective approaches to attention-based neural machine translation". In: *arXiv preprint arXiv:1508.04025*.

Ma, Zhuang, Yichao Lu, and Dean Foster (2015). "Finding linear structure in large datasets with scalable canonical correlation analysis". In: *International Conference on Machine Learning*, pp. 169–178.

Martin, Travis, Jake M Hofman, Amit Sharma, Ashton Anderson, and Duncan J Watts (2016). "Exploring limits to prediction in complex social systems". In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 683–694.

Masci, Jonathan, Michael M Bronstein, Alexander M Bronstein, and Jürgen Schmidhuber (2014). "Multimodal similarity-preserving hashing". In: *IEEE transactions on pattern analysis and machine intelligence* 36.4, pp. 824–830.

McAuley, Julian and Alex Yang (2016). "Addressing complex and subjective product-related queries with customer reviews". In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 625–635.

Mcauliffe, J. D. and D. M. Blei (2008). "Supervised topic models". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 121–128.

McKee, Rebecca (2013). "Ethical issues in using social media for health and health care research". In: *Health Policy* 110.2, pp. 298–301.

Mcnamee, Paul and James Mayfield (2004). "Character n-gram tokenization for European language text retrieval". In: *Information retrieval* 7.1-2, pp. 73–97.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013a). "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*, pp. 3111–3119.

— (2013b). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems (NIPS)*, pp. 3111–3119.

Miller, George A and Walter G Charles (1991). "Contextual correlates of semantic similarity". In: *Language and cognitive processes* 6.1, pp. 1–28.

Mimno, D. and A. McCallum (2008). "Topic models conditioned on arbitrary features with Dirichlet-multinomial regression". In: *UAI*.

Mitchell, Margaret, Kristy Hollingshead, and Glen Coppersmith (2015). "Quantifying the Language of Schizophrenia in Social Media". In: *Proceedings of the 2nd*

Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality. Denver, Colorado: Association for Computational Linguistics, pp. 11–20. URL: http://www.aclweb.org/anthology/W15-1202.

Mohammad, Saif M., Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry (2016). "Semeval-2016 Task 6: Detecting Stance in Tweets". In: *Proceedings of the International Workshop on Semantic Evaluation*. SemEval '16. San Diego, California.

Mroueh, Youssef, Etienne Marcheret, and Vaibhava Goel (2015). "Asymmetrically Weighted CCA And Hierarchical Kernel Sentence Embedding For Multimodal Retrieval". In: *arXiv preprint arXiv:1511.06267*.

Murakami, Akiko and Rudy Raymond (2010). "Support or oppose?: classifying positions in online debates from reply activities and opinion expressions". In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, pp. 869–875.

Nguyen, Dong, Noah A Smith, and Carolyn P Rosé (2011). "Author age prediction from text using linear regression". In: *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. Association for Computational Linguistics, pp. 115–123.

Nguyen, Dong, Rilana Gravel, Dolf Trieschnigg, and Theo Meder (2013). ""How Old Do You Think I Am?" A Study of Language and Age in Twitter." In: *ICWSM*.

Nguyen, Dong, Dolf Trieschnigg, A. Seza Dogruöz, Rilana Gravel, Mariet Theune, Theo Meder, and Franciska De Jong (2014). "Predicting Author Gender and Age from Tweets: Sociolinguistic Theories and Crowd Wisdom". In: *Proceedings of COLING 2014*.

Niu, Jinghao, Yehui Yang, Siheng Zhang, Zhengya Sun, and Wensheng Zhang (2018). "Multi-task Character-Level Attentional Networks for Medical Concept Normalization". In: *Neural Processing Letters*, pp. 1–18.

O'Connor, Brendan, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith (2010b). "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series". In: *Proceedings of the International AAAI Conference on Weblogs and Social Media*.

O'Connor, Brendan, Ramnath Balasubramanyan, Bryan R Routledge, Noah A Smith, et al. (2010a). "From tweets to polls: Linking text sentiment to public opinion time series." In: *Icwsm* 11.122-129, pp. 1–2.

Padrez, Kevin A, Lyle Ungar, Hansen Andrew Schwartz, Robert J Smith, Shawndra Hill, Tadas Antanavicius, Dana M Brown, Patrick Crutchley, David A Asch, and Raina M Merchant (2015). "Linking social media and medical record data: a study

of adults presenting to an academic, urban emergency department". In: *BMJ Qual Saf*, bmjqs–2015.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs up?: sentiment classification using machine learning techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pp. 79–86.

Pang, Bo, Lillian Lee, et al. (2008). "Opinion mining and sentiment analysis". In: *Foundations and Trends® in Information Retrieval* 2.1–2, pp. 1–135.

Park, Greg, H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, David J Stillwell, Michal Kosinski, Lyle H Ungar, and Martin EP Seligman (2015). "Automatic personality assessment through social media language". In: *Journal of Personality and Social Psychology*.

Paul, M. and R. Girju (2009). "Cross-Cultural Analysis of Blogs and Forums with Mixed-Collection Topic Models". In: *EMNLP*, pp. 1408–1417. URL: http://www.aclweb.org/anthology/D/D09/D09-1146.pdf.

Paul, Michael J and Mark Dredze (2013). "Drug Extraction from the Web: Summarizing Drug Experiences with Multi-Dimensional Topic Models." In: *HLT-NAACL*, pp. 168–178.

— (2015). "SPRITE: Generalizing topic models with structured priors". In: *Transactions of the Association for Computational Linguistics* 3, pp. 43–57.

— (2011). "You are what you Tweet: Analyzing Twitter for public health." In: *Icwsm* 20, pp. 265–272.

Paul, Michael John (2015a). "Topic Modeling with Structured Priors for Text-Driven Science". PhD thesis. Johns Hopkins University.

— (2015b). "Topic Modeling with Structured Priors for Text-Driven Science". PhD thesis. Johns Hopkins University. Chap. 2, pp. 9–55.

Pedersen, Ted (2015). "Screening Twitter Users for Depression and PTSD with Lexical Decision Lists". In: *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Denver, Colorado: Association for Computational Linguistics, pp. 46–53. DOI: 10.3115/v1/W15-1206. URL: http://aclweb.org/anthology/W15-1206.

Pennacchiotti, Marco and Ana-Maria Popescu (2011). "A Machine Learning Approach to Twitter User Classification." In: *Icwsm* 11.1, pp. 281–288.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing*. Vol. 14, pp. 1532–1543.

Phan, X., L. Nguyen, and S. Horiguchi (2008). "Learning to classify short and sparse text & web with hidden topics from large-scale data collections". In: *WWW '08: Proceeding of the 17th international conference on World Wide Web*. Beijing, China: ACM, pp. 91–100.

Phelan, Owen, Kevin McCarthy, and Barry Smyth (2009). "Using twitter to recommend real-time topical news". In: *Proceedings of the third ACM conference on Recommender systems*. ACM, pp. 385–388.

Plank, Barbara and Dirk Hovy (2015). "Personality traits on twitter—or—how to get 1,500 personality tests in a week". In: *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 92–98.

Preoţiuc-Pietro, Daniel, Johannes Eichstaedt, Gregory Park, Maarten Sap, Laura Smith, Victoria Tobolsky, Hansen Andrew Schwartz, and Lyle H Ungar (2015). "The Role of Personality, Age and Gender in Tweeting about Mental Illnesses". In: *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. NAACL.

Preoţiuc-Pietro, Daniel and Lyle Ungar (2018). "User-level Race and Ethnicity Predictors from Twitter Text". In: *COLING*.

Preotiuc-Pietro, Daniel, Vasileios Lampos, and Nikolaos Aletras (2015). "An analysis of the user occupational class through Twitter content". In: *ACL*.

Purohit, Hemant, Yiye Ruan, Amruta Joshi, Srinivasan Parthasarathy, and Amit Sheth (2011). "Understanding user-community engagement by multi-faceted features: A case study on twitter". In: *WWW Workshop on Social Media Engagement (SoME)*.

Rahimi, Afshin, Trevor Cohn, and Tim Baldwin (2018). "Semi-supervised User Geolocation via Graph Convolutional Networks". In: *Proceedings of the 2018 Conference of the Association for Computational Linguistics (ACL)*.

Rajendran, Janarthanan, Mitesh M Khapra, Sarath Chandar, and Balaraman Ravindran (2015). "Bridge Correlational Neural Networks for Multilingual Multimodal Representation Learning". In: *arXiv preprint arXiv:1510.03519*.

Ramage, D., D. Hall, R. Nallapati, and C.D. Manning (2009). "Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora". In: *EMNLP*. URL: http://dl.acm.org/citation.cfm?id=1699510.1699543.

Ramage, Daniel, Susan T. Dumais, and Daniel J. Liebling (2010). "Characterizing Microblogs with Topic Models". In: *ICWSM*.

Rao, Delip, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta (2010). "Classifying latent user attributes in twitter". In: *Proceedings of the 2nd international workshop on Search and mining user-generated contents*. ACM, pp. 37–44.

Rastogi, Pushpendre, Benjamin Van Durme, and Raman Arora (2015). "Multiview LSA: Representation learning via generalized CCA". In: *North American Association for Computational Linguistics (NAACL)*.

Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). "SemEval-2017 task 4: Sentiment analysis in Twitter". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 502–518.

Rosenthal, Sara and Kathleen McKeown (2011). "Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pp. 763–772.

Rubenstein, Herbert and John B Goodenough (1965). "Contextual correlates of synonymy". In: *Communications of the ACM* 8.10, pp. 627–633.

Rush, Alexander M, David Sontag, Michael Collins, and Tommi Jaakkola (2010). "On dual decomposition and linear programming relaxations for natural language processing". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1–11.

Ruths, Derek and Jürgen Pfeffer (2014). "Social media for large studies of behavior". In: *Science* 346.6213, pp. 1063–1064.

Salehi, Bahar, Fei Liu, Timothy Baldwin, and Wilson Wong (2018). "Multitask Learning for Query Segmentation in Job Search". In: *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, pp. 179–182.

Sandhaus, Evan (2008). "The new york times annotated corpus". In: *Linguistic Data Consortium, Philadelphia* 6.12, e26752.

Sarawgi, Ruchita, Kailash Gajulapalli, and Yejin Choi (2011). "Gender attribution: tracing stylometric evidence beyond topic and genre". In: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 78–86.

Schulz, Claudia, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych (2018). "Multi-Task Learning for Argumentation Mining". In: *arXiv preprint arXiv:1804.04083*.

Schwartz, Andrew H., Johannes Eichstaedt, L. Margaret Kern, Gregory Park, Maarten Sap, David Stillwell, Michal Kosinski, and Lyle Ungar (2014). "Towards Assessing Changes in Degree of Depression through Facebook". In: *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Baltimore, Maryland, USA: Association for

Computational Linguistics, pp. 118–125. DOI: 10.3115/v1/W14-3214. URL: http://aclweb.org/anthology/W14-3214.

Schwartz, Hansen Andrew, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. (2013a). "Personality, gender, and age in the language of social media: The open-vocabulary approach". In: *PloS one* 8.9.

Schwartz, Hansen Andrew, Johannes C Eichstaedt, Lukasz Dziurzynski, Margaret L Kern, Eduardo Blanco, Michal Kosinski, David Stillwell, Martin EP Seligman, and Lyle H Ungar (2013b). "Toward Personality Insights from Language Exploration in Social Media." In: *AAAI Spring Symposium: Analyzing Microtext*.

She, Jieying and Lei Chen (2014). "Tomoha: Topic model-based hashtag recommendation on twitter". In: *International conference on World wide web (WWW)*. International World Wide Web Conferences Steering Committee, pp. 371–372.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034*.

Smith, Wendell R (1956). "Product differentiation and market segmentation as alternative marketing strategies". In: *Journal of marketing* 21.1, pp. 3–8.

Snoek, Jasper, Hugo Larochelle, and Ryan P Adams (2012). "Practical Bayesian optimization of machine learning algorithms". In: *Advances in Neural Information Processing Systems*, pp. 2951–2959.

Søgaard, Anders and Yoav Goldberg (2016). "Deep multi-task learning with low level tasks supervised at lower layers". In: *The 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, p. 231.

Somasundaran, Swapna and Janyce Wiebe (2009). "Recognizing stances in online debates". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, pp. 226–234.

Stefanone, Michael A, Gregory D Saxton, Michael J Egnoto, Wayne Wei, and Yun Fu (2015). "Image attributes and diffusion via twitter: The case of# guncontrol". In: *System Sciences (HICSS), 2015 48th Hawaii International Conference on*. IEEE, pp. 1788–1797.

Strauss, Benjamin, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu (2016). "Results of the wnut16 named entity recognition shared task". In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pp. 138–144.

Sutton, Charles, Andrew McCallum, and Khashayar Rohanimanesh (2007). "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data". In: *Journal of Machine Learning Research* 8.Mar, pp. 693–723.

Tan, Chenhao, Lillian Lee, and Bo Pang (2014). "The effect of wording on message propagation: Topic-and author-controlled natural experiments on Twitter". In: *arXiv preprint arXiv:1405.1438*.

Tang, Duyu, Bing Qin, and Ting Liu (2015). "Document modeling with gated recurrent neural network for sentiment classification". In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432.

Tao, Wang and Liu Yang (2017). "Multiview Community Discovery Algorithm via Nonnegative Factorization Matrix in Heterogeneous Networks". In: *Mathematical Problems in Engineering* 2017.

Tausczik, Yla R and James W Pennebaker (2010). "The psychological meaning of words: LIWC and computerized text analysis methods". In: *Journal of language and social psychology* 29.1, pp. 24–54.

Thacker, Stephen B and Ruth L Berkelman (1988). "Public health surveillance in the United States". In: *Epidemiologic reviews* 10, pp. 164–90.

Tran, Tung and Ramakanth Kavuluru (2017). "Predicting mental conditions based on "history of present illness" in psychiatric notes with deep neural networks". In: *Journal of biomedical informatics* 75, S138–S148.

Tsai, Chun-Yu and John R Kender (2017). "Detecting Culture-specific Tags for News Videos through Multimodal Embedding". In: *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. ACM, pp. 68–74.

Tumasjan, Andranik, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe (2010). "Predicting elections with twitter: What 140 characters reveal about political sentiment." In: *Icwsm* 10.1, pp. 178–185.

Uurtio, Viivi, João M Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu (2017). "A tutorial on canonical correlation methods". In: *ACM Computing Surveys (CSUR)* 50.6, p. 95.

Van De Velden, Michel and Tammo HA Bijmolt (2006). "Generalized canonical correlation analysis of matrices with missing rows: a simulation study". In: *Psychometrika* 71.2, pp. 323–331.

Volkova, Svitlana (2015). "Predicting Demographics and Affect in Social Networks". PhD thesis. Johns Hopkins University.

Volkova, Svitlana, Glen Coppersmith, and Benjamin Van Durme (2014a). "Inferring User Political Preferences from Streaming Communications." In: *ACL*, pp. 186–196.

— (2014b). “Inferring User Political Preferences from Streaming Communications.” In: *Association for Computational Linguistics (ACL)*, pp. 186–196.

Volkova, Svitlana and Benjamin Van Durme (2015). “Online Bayesian Models for Personal Analytics in Social Media.” In:

Volkova, Svitlana, Theresa Wilson, and David Yarowsky (2013). “Exploring Demographic Language Variations to Improve Multilingual Sentiment Analysis in Social Media.” In: *Proceedings of EMNLP*, pp. 1815–1827.

Volkova, Svitlana, Yoram Bachrach, Michael Armstrong, and Vijay Sharma (2015a). “Inferring Latent User Properties from Texts Published in Social Media.” In: *AAAI*, pp. 4296–4297.

— (2015b). “Inferring Latent User Properties from Texts Published in Social Media.” In: *AAAI*, pp. 4296–4297.

Vosoughi, Soroush, Prashanth Vijayaraghavan, and Deb Roy (2016). “Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder”. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pp. 1041–1044.

Wallach, Hanna M, Iain Murray, Ruslan Salakhutdinov, and David Mimno (2009). “Evaluation methods for topic models”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, pp. 1105–1112.

Wang, Weiran, Raman Arora, Karen Livescu, and Jeff Bilmes (2015). “On deep multi-view representation learning”. In: *International Conference on Machine Learning*, pp. 1083–1092.

Wang, Xuerui, Wei Li, Ying Cui, Ruofei Zhang, and Jianchang Mao (2011). “Click-through rate estimation for rare events in online advertising”. In: *Online Multimedia Advertising: Techniques and Technologies*. IGI Global, pp. 1–12.

Westbury, John R. (1994). “X-Ray Microbeam Speech Production Database User’s Handbook”. In: *Waisman Center on Mental Retardation & Human Development University of Wisconsin Madison, WI 53705-2280*.

WHO, World Health Organization (2016). *Gender and Women’s Mental Health*. `http://www.who.int/mental_health/prevention/genderwomen/en/`.

Wu, Shaomei, Jake M Hofman, Winter A Mason, and Duncan J Watts (2011). “Who says what to whom on twitter”. In: *Proceedings of the 20th international conference on World wide web*. ACM, pp. 705–714.

Xing, Linzi and Michael J Paul (2017). “Incorporating Metadata into Content-Based User Embeddings”. In: *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 45–49.

Yan, Rui, Mirella Lapata, and Xiaoming Li (2012). "Tweet recommendation with graph co-ranking". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pp. 516–525.

Yang, Min, Wenting Tu, Jingxuan Wang, Fei Xu, and Xiaojun Chen (2017). "Attention Based LSTM for Target Dependent Sentiment Classification." In: *AAAI*, pp. 5013–5014.

Yang, Yi and Jacob Eisenstein (2017). "Overcoming Language Variation in Sentiment Analysis with Social Attention". In: *Transactions of the Association of Computational Linguistics* 5.1, pp. 295–307.

Yano, T., W. Cohen, and N. Smith (2009). "Predicting Response to Political Blog Posts with Topic Models". In: *The 7th Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. Boulder, CO, USA.

Yazdavar, Amir Hossein, Hussein S Al-Olimat, Monireh Ebrahimi, Goonmeet Bajaj, Tanvi Banerjee, Krishnaprasad Thirunarayan, Jyotishman Pathak, and Amit Sheth (2017). "Semi-supervised approach to monitoring clinical depressive symptoms in social media". In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ACM, pp. 1191–1198.

Yu, Kai, Volker Tresp, and Anton Schwaighofer (2005). "Learning Gaussian processes from multiple tasks". In: *Proceedings of the 22nd international conference on Machine learning*. ACM, pp. 1012–1019.

Yu, Mo, Matthew R Gormley, and Mark Dredze (2015). "Combining Word Embeddings and Feature Embeddings for Fine-grained Relation Extraction". In: *Proceedings of the 14th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1374–1379.

Zangerle, Eva, Wolfgang Gassler, and Günther Specht (2013). "On the impact of text similarity functions on hashtag recommendations in microblogging environments". In: *Social Network Analysis and Mining* 3.4, pp. 889–898.

Zarrella, Guido and Amy Marsh (2016). "MITRE at semeval-2016 task 6: Transfer learning for stance detection". In: *arXiv preprint arXiv:1606.03784*.

Zeiler, Matthew D (2012). "ADADELTA: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701*.

Zhang, Yu and Dit Yan Yeung (2011). "Multi-task learning in heterogeneous feature spaces". In: *25th AAAI Conference on Artificial Intelligence and the 23rd Innovative Applications of Artificial Intelligence Conference, AAAI-11/IAAI-11, San Francisco, CA, United States, 7-11 August 2011, Code 87049, Proceedings of the National Conference on Artificial Intelligence*, p. 574.

Zhu, X., D. Blei, and J. Lafferty (2006). *TagLDA: bringing document structure knowledge into topic models*. Tech. rep. Technical Report TR-1553. University of Wisconsin.

Zou, Bin, Vasileios Lampos, and Ingemar Cox (2018). "Multi-Task Learning Improves Disease Models from Web Search". In: *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, pp. 87–96.

Vita

Adrian Benton received the B.A. degree in Linguistics from the University of Pennsylvania in 2008. He received the M.S. degree in Computer Science from the University of Pennsylvania in 2012, and enrolled in the Computer Science Ph.D. program at Johns Hopkins University in 2013. His research centers around applying machine learning techniques to analyze social media data.