Accelerating Text-as-Data Research in Computational Social Science

Dallas Card

August 2019
CMU-ML-19-109

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Noah A. Smith, Chair
Artur Dubrawski
Geoff Gordon
Dan Jurafsky (Stanford University)

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

This work was supported by a Natural Sciences and Engineering Research Council of Canada Postgraduate Scholarship, NSF grant IIS-1211277, an REU supplement to NSF grant IIS-1562364, a Bloomberg Data Science Research Grant, a University of Washington Innovation Award, and computing resources provided by XSEDE.

Keywords: machine learning, natural language processing, computational social science, graphical models, interpretability, calibration, conformal methods

Abstract

Natural language corpora are phenomenally rich resources for learning about people and society, and have long been used as such by various disciplines such as history and political science. Recent advances in machine learning and natural language processing are creating remarkable new possibilities for how scholars might analyze such corpora, but working with textual data brings its own unique challenges, and much of the research in computer science may not align with the desiderata of social scientists. In this thesis, I present a line of work on developing methods for computational social science focused primarily on observational research using natural language text. Throughout, I take seriously the concerns and priorities of the social sciences, leading to a focus on aspects of machine learning which are otherwise sometimes secondary, including calibration, interpretability, and transparency. Two ideas which unify this work are the problems of *exploration* and *measurement*, and as a running example I consider the problem of analyzing how news sources frame contemporary political issues. Following the introduction, I devote one chapter to providing the necessary background on computational social science, framing, and the "text as data" paradigm. Subsequent chapters each focus on a particular model or method that strives to address some aspect of research which may be of particular interest to social scientists. Chapters 3 and 4 focus on the unsupervised setting, with the former presenting a model for learning archetypal character representations, and the latter presenting a framework

for neural document models which can flexibly incorporate metadata. Chapters 5 and 6 focus on the supervised setting and present alternately, a method for measuring label proportions in text in the presence of domain shift, and a variation on deep learning classifiers which produces more transparent and robust predictions. The final chapter concludes with implications for computational social science and possible directions for future work.

Contents

Acknowledgements

Writing an acknowledgements section is a powerful reminder that scholarship is a collective endeavour, and that we all depend on, and benefit so much from, the contributions of countless others, both personally and professionally. I certainly could not have completed this work without the help of innumerable colleagues, friends, and family members.

First and foremost, I would like to thank my advisor, Noah Smith, for his unbounded support, encouragement, and insight, and for creating an exceptional learning environment in which so many are able to thrive while continuing to challenge both themselves and each other. In addition, I wish to express my sincere thanks to all members of my thesis committee, my teachers, collaborators, and co-authors who have taught me so much, all current and former members of Noah's ARK for their camaraderie and mentorship, the provocative interlocutors at the Tech Policy Lab for their stimulating conversation, all of my friends who have kept me sane and helped me to grow, the incredible support staff of the Machine Learning Department at Carnegie Mellon University, as well as everyone at the Paul G. Allen School at the University of Washington for providing me with a temporary second home.

While far from exhaustive, I would like to express a special gratitude to Alnur Ali, Waleed Ammar, David Bamman, Antoine Bosselut, Amber Boydstun, Ben Cowley, Tam Dang, Jesse Dodge, Artur Dubrawski, Chris Dyer, Nicholas FitzGerald, Alona Fyshe,

Emily Kalah Gade, Geoff Gordon, Justin Gross, Suchin Gururangan, Ari Holtzman, George Ibrahim, Dan Jurafsky, Lingpeng Kong, Will Lowe, Bill McDowell, Darren McKee, Tom Mitchell, Jared Moore, Brendan O'Connor, Anthony Platanios, Aaditya Ramdas, Philip Resnik, Maarten Sap, Nathan Schneider, Dan Schwartz, Roy Schwartz, Swabha Swayamdipta, Yanchuan Sim, Diane Stidle, Chenhao Tan, Margot Taylor, Sam Thompson, Ryan Tibshirani, Leron Vandsburger, Hanna Wallach, Jing Xiang, Mark Yatskar, Dani Yogatama, and Michael Zhang. Above all, I would like to thank my family, especially my parents, for their love and support, and for giving me the confidence and curiosity to continue exploring and follow this path, wherever it might lead.

Chapter 1

Introduction

Over the past decade, the field of machine learning has grown massively in prominence and importance, influencing many neighboring academic disciplines, and becoming the de facto core toolkit of modern data analysis (Jordan and Mitchell, 2015). Through creative innovations in algorithms, elegant theoretical foundations, and effective implementations, research in machine learning has demonstrated that it is possible to derive insights from, and make accurate predictions about, even very complicated and large-scale datasets, above and beyond what was thought to be achievable with traditional statistics.

Among the fields eager to make use of these innovations is the diverse set of disciplines we think of as the social sciences. These disciplines embody a rich history of trying to make sense of the behavior of individuals, communities, and society. Many methods have been developed over the years in pursuit of this objective, but most recently there has been a flourishing of research under the banner of *computational social science*, in which ideas and methods from computer science are being integrated into the process of studying people and their interactions (Lazer et al., 2009). In addition to advances in methodology, this work has been fueled, in part, by huge increases

in the amount of data, including text, that people generate, both actively and passively, as they go about their lives (Salganik, 2017).

Text is a particularly rich source of potential insight, as it can simultaneously represent both aggregate trends, such as changes in language use over time, as well as more individual expressions of what people think, believe, and wish to communicate (O'Connor et al., 2011). However, there are several ways in which the needs of social science investigations often differ from conventional prediction problems, including in their goals, priorities, and criteria for success (Hopkins and King, 2010; Wallach, 2018). Further complicating matters, text data presents unique difficulties for machine learning, and this is especially true in social science applications, where insight and theory are often prized above and beyond raw predictive power (Grimmer and Stewart, 2013).

In this thesis, I bring together a line of work on developing methods in machine learning and natural language processing attuned to the needs of computational social science and so-called "text as data" research. Some of this this work involves conventional model building in both supervised and unsupervised settings. However, I try throughout to take seriously the priorities of the social sciences, as well as the necessity of collaborative efforts. I firmly believe that the most interesting results emerge from interdisciplinary teams which bring together people with diverse expertise. As such, the goal should be not merely to develop tools that can be used by people in other disciplines, but to create the framework in which teams can come together and actively participate in all aspects of the research cycle.

In particular, I draw inspiration from the needs and priorities of scholars in other domains, such as political science, leading me to focus on otherwise somewhat secondary aspects of machine learning, including iterative modeling, interpretability, transparency, calibration, and credibility. However, far from being parochial concerns,

most of these are in fact important to all scientific investigations, and I believe that machine learning as a field can itself learn from the best practices of the social sciences.

Although the focus of this thesis is on methods, not on substantive sociological findings, as a running example threaded throughout this thesis, I consider the issue of *framing* — that is, the idea that the way in which we present information can make a difference (Gitlin, 1980; Entman, 1993; Kahneman, 2011). While framing remains a challenging idea to study, it is a useful example of the type of complex phenomenon that social scientists wish to discover and measure in text (see §2.4).

1.1 Thesis statement

In this thesis, I argue that there is great potential for machine learning and natural language processing to be useful in social science research, but that the priorities of the social sciences necessitate placing greater emphasis on sometimes secondary aspects of computational methods, including interpretability, transparency, calibration, reliability, and cost. I illustrate this by developing a variety of different models and methods, each of which illustrates one or more of these aspects that is sometimes overlooked, driven by the priorities of researchers in the social sciences. In particular, I am especially focused on the problems of exploration and measurement—that is, how can we construct useful machine learning models which allow us to a) make sense of large text corpora; and b) convert rich data, such as text, into quantitative measurements of theoretical concepts of interest.

The focus of this thesis is on interdisciplinary research in the social sciences, but it can also be seen as part of a more general data science paradigm, one that is particularly attuned to text data. Although there is an important role for experiments in social science, I emphasize observational over experimental work. Similarly, while there are

often good reasons to develop unique models for particular applications, I am mostly interested in collaborative scenarios, in which people from other disciplines are able to make greater use of ideas and tools from machine learning and natural language processing.

1.2 Structure of this thesis

The remainder of this thesis is made up of five chapters. Chapter 2 provides the relevant background on computational social science and "text as data", a summary of desiderata for social science methods, and an overview of work on framing that is used as a running example throughout the thesis. Each of the remaining chapters proposes a model or method, each of which attempts to address at least one concern that is of particular importance to social scientists.

Chapters 3 and 4 deal with the unsupervised setting. The first of these proposes a specific model for unsupervised learning of archetypal character representations. The latter proposes a model for documents with metadata, emphasizing the potential of neural variational inference to allow for model customization without model-specific derivations. Chapters 5 and 6, by contrast focus on the supervised setting. The first of these chapters focuses specifically on text classification as a tool for measurement in the presence of domain shift. The last chapter is not focused exclusively on text, but instead shows how a small change to any deep learning classifier can produce predictions that are more transparent and robust, particularly for out-of-domain data.

I conclude with a summary of implications for computational social science and ideas about possible directions for future work. Taken together, this collection of chapters can be understood as an effort to push the field towards a more nuanced approach to interdisciplinary applications of machine learning.

Chapter 2

Background

2.1 Computational social science and "text as data"

Despite a growing trend towards interdisciplinary research, computer scientists, social scientists, and policy makers often have different goals, methods, and criteria for success (Hopkins and King, 2010; O'Connor et al., 2011; Grimmer and Stewart, 2013; Kleinberg et al., 2015; Wallach, 2016, 2018). Broadly speaking, social science proceeds by theorizing causal explanations of social phenomena, and evaluating, interrogating, and refining those theories. While in many cases this endeavor proceeds through constructing models and testing hypotheses in ways that will be familiar to machine learning researchers (either through experimental or observational studies), there is also a large amount of effort devoted to exploratory work, much of which focuses on providing "thick" description and context, which may be used for subsequent theorizing.[1]

[1] Ideally, a good theory should be able to make useful predictions, and comparing predictions against reality is an important form of evaluation; however, because of the complexity inherent in social systems, theories in social science are sometimes more useful as sources of insight, or as a way of thinking about a new phenomenon, even if they have limited predictive power.

Although every project is different, a large amount of research in social science disciplines can be broken down into the following stages: i) formulating research questions based on existing theory; ii) collecting and annotating data; iii) exploratory data analysis; iv) building, testing, and applying models; v) interpretation and visualization of results; vi) refining theoretical ideas for further investigation. This is clearly not a unidirectional process, but rather one in which the results at each stage inform decision making at the previous stages on the next iteration, as illustrated in Figure 2.1.[2]

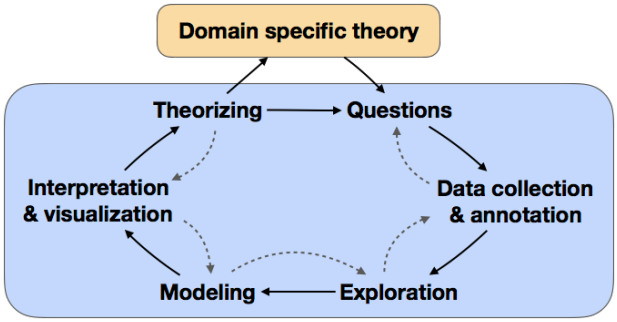


Figure 2.1: Schematic depiction of the interaction between existing theory for a given domain and the research cycle for a particular project. Feedback loops (dashed arrows) emphasize how each stage is informed by insights from later stages.

While the research cycle described above applies to a great deal of traditional research, there is increasingly a movement within the social sciences to make use of digital data, online experiments, machine learning, and other computational methods, hence the idea of *computational social science* (Lazer et al., 2009; O'Connor et al., 2011; Mason et al., 2014; Salganik, 2017; Wallach, 2018). Different researchers may assign different meanings to this term, but the general idea is that computational social science is a collaborative area of research in which insights and methods from computer science (e.g., large-scale statistical inference, network analysis, text analysis, etc.) are brought to bear on asking and answering questions about society.

As the name would suggest, computational social science is inherently interdisci-

[2] David Blei provides a similar description of the iterative nature of latent variable modeling in his presentation of "Box's loop" (Blei, 2014).

plinary. Not only do scholars from the social sciences bring background knowledge and domain expertise that is essential in formulating research questions, they are often in the best position to evaluate whether the results of an investigation are sensible or meaningful in light of existing theory. Unfortunately, there tends to be a trade-off between familiar, interpretable, and relatively simple models (such as linear and logistic regression), and modern machine learning models, which may be more powerful, but also more difficult to understand or interpret. Nevertheless, there is much greater scope for enabling researchers to bring implicit or explicit knowledge to bear on the other stages, by expanding the range of tools and methods that are readily available to those with less expertise in machine learning or natural language processing.

While a great deal of computational social science deals with traditional types of data, such as opinion polls or voting behavior, a growing community of scholars in computational social science is especially interested in the potential of using written *text* as a source of insight. The phrase "text as data" may seem redundant to researchers in natural language processing, but it has been a compelling idea in recent years, providing the name for a research association, an annual meeting, and a widely cited summary paper (Grimmer and Stewart, 2013).

Textual archives clearly play a fundamental role in certain sub-fields of the social sciences and humanities; what is novel is the idea of considering raw unstructured text as data which can be interpreted quantitatively using automated or semi-automated methods. In addition, the phrase suggests an allusion to "big data", underscoring that there is an enormous amount of unstructured text available that has been written by people; as with the many other electronic traces we leave in our lives, this sort of "found data" has the potential to be a remarkable source of insight into society, in some cases with the ability to answer questions that have little to do with the original reasons for the existence of such data.

2.2 Exploration and measurement

Compared with more directly quantitative data, such as images or structured records, text presents unique challenges for statistical analysis, yet also offers unique prospects in terms of the potential for answering social scientific questions and learning about society more broadly. Developments in statistical and computational methods for text analysis over the past decade have vastly expanded the range of what is possible, but there continues to be a gap between the potential demonstrated in select research projects, and the range of tools and methods that have seen widespread adoption in the broader research community (Grimmer and Stewart, 2013; O'Connor, 2014).

When working with massive text corpora, there are two basic ways of using computational methods that are especially important (O'Connor et al., 2011). The first is *exploration*, in which we want to make sense of and discover things about a corpus of documents that is too large to be read.[3] The second is to use computational methods as tools for *measurement*, ones which allow us to convert documents into quantitative measurements of social constructs. Although distinct, these two approaches are clearly related, as corpus exploration typically involves quantifying text in ways which may suggest potential measurements.

Broadly speaking, there are a few different ways of approaching the problem of making measurements from text. The traditional approach used in many social sciences is that of *content analysis* (Krippendorff, 2012). Given a set of possible *codes*, each of which typically represents a concept related to a particular question of interest, human annotators read the text and assign codes to documents or parts of documents. These codes are typically developed in an iterative fashion (and might be unique to each

[3]The term "document" will be used as a stand-in for any appropriately sized piece of text, even if it exists only in virtual form. In the context of this thesis, a document could be a newspaper article, a tweet, an online product review, etc.

project), and are formalized in a codebook. Annotating large numbers of documents thus allows researchers to make inferences or draw conclusions about a corpus that relates to the substantive question of interest. While trained annotators can typically identify these codes in text with some acceptable level of agreement, this remains a somewhat subjective process, and of course does not scale beyond the limits of human labour.

Alternatively, supervised learning provides a way to try to augment the coding process, which I will consider in more detail in Chapter 5. In principle, predicted labels can serve as a supplement to human-coded documents, though this raises additional questions about validity, reliability, and reproducibility (see §2.3). However, because of the richness and ambiguity of human language, this remains a challenging problem. Recent developments in pretraining have brought enormous gains in tasks such as text classification (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019), but more complicated deep models have certain downsides, including opaqueness, over-confidence, and cost (see Chapter 6; also Gururangan et al., 2019).

Finally, it is also possible to use unsupervised methods to attempt to measure relevant concepts or categories, often by incorporating prior knowledge. While unsupervised learning is typically thought of as a tool for exploration (and indeed is often useful for that purpose), Wallach (2016) emphasizes that it is also a natural way to approach the measurement problem. Specifically, by defining a generative model, and using the available data to make inferences about latent variables, we can obtain posterior estimates which can serve as the basis of measurements (Blei, 2014). Topic models such as latent Dirichlet allocation (Blei et al., 2003) are the most familiar example (see also Boyd-Graber et al., 2017), but it is also possible to develop more specialized models, as I do in Chapter 3 for *personas*, as well as more broadly applicable models, as I do in Chapter 4 for documents with metadata. Although evaluation remains a challenge for

unsupervised learning, the probabilistic graphical models framework allows for rich specification of prior knowledge and can, in some cases, allow us to attribute specific interpretations to various parts of a model.

2.3 Desiderata in computational social science methods

Although recent work in machine learning *fairness* has drawn attention to some of the trade-offs involved (Hardt et al., 2016; Kleinberg et al., 2017b), the conventional objective in most machine learning research is still that of maximizing accuracy. Treating prediction and inference as types of measurement, by contrast, requires taking seriously the rich body of work that is prominent within the social sciences concerning possible threats to validity. In this section, I review some of the important considerations for methods used in the social sciences. While most of these are broadly applicable, I specifically have in mind the application of making measurements of text, and provide examples below.

A complete discussion of measurement is beyond the scope of this thesis, but in general, the following considerations are particularly important:

- **Validity**: Validity is a central concept in the discussion of research methods in social science, and includes multiple different aspects (Drost, 2011; Bhattacherjee, 2012; Nguyen et al., 2016). Some of these, such as *statistical conclusion validity*, *internal validity*, and *external validity* have to do with drawing conclusions from measurements (i.e., is the relationship statistically significant and robust? Is it causal? Will it generalize?). Most relevant here, however, is the notion of *construct validity*. In social science research, one typically assumes the existence of a theoretical construct which is theorized to have some relevance to a question, and which one would ideally like to measure, such as "political ideology". In

practice, however, one will typically need to rely on some sort of instrument which will measure an operationalized form of the construct (i.e., something that can be explicitly measured). Construct validity is an assessment of how well the thing that we can measure actually represents the construct that we care about. Discussions of research methods (e.g., Bhattacherjee, 2012) typically break this down into many sub-components, such as *face validity* (is it reasonable "on-its-face"?), *content validity* (does it completely and exclusively measure the construct of interest), and *convergent validity* (does it agree with existing measure of the construct?). Ultimately, however, it is difficult to ever conclusively prove validity; rather, one should be aware of various common failure modes, and validate to the extent possible.

- **Reliability**: Even if one has a valid way of measuring a particular construct (for example, by using human annotators), it is still necessary to consider possible measurement error, just as we would for measuring a physical property, such as temperature. As in all measurements, we should ask about both random and systematic errors. For a given instrument, we can think about systematic errors as a type of bias, and random errors as variance. An ideal instrument would be unbiased with low variance, but in practice we will likely have to worry about both (Bhattacherjee, 2012). Of course, in many settings, it is preferable to introduce some bias in order to obtain lower variance, but unbiasedness is nevertheless prioritized in some domains. For categorical measurements, it is more intuitive to discuss error in terms of properties of the confusion matrix, such as accuracy, false positive rate, or sensitivity. Moreover, for probabilistic classifiers, it is also useful to assess calibration (the long run correctness of predicted probabilities, as discussed in Chapter 5). Note, however, that a single number is not sufficient to summarize all properties on an instrument, even for a binary classifier.

- **Reproducibility**: While reliability typically refers to the use of a particular instrument, reproducibility refers to the broader ability of other researchers to reproduce a particular measurement or result (Gundersen and Kjensmo, 2018). Although reproducibility gets less attention within discussions of research methods in social science, compared to validity and reliability, it has become a growing priority with the rise of the so-called "replication crisis" (Goodman et al., 2016). While reproducibility involves technical considerations, it also relates to how easily and effectively methodologies can be communicated and understood; simpler methods are likely to be more reproducible, though publishing code, for example, can enhance the reproducibility of any method. Some approaches to machine learning naturally facilitate reproducibility (such as when using convex optimization under controlled conditions). However, with the increasing dominance of deep neural networks, there are numerous hurdles to this sort of reproducibility, including the impact of large numbers of hyperparameters, random seeds, software dependencies, hardware differences, and so on. Reproducibility has also received attention within the natural language processing and text-as-data communities (Radev, 2009; Dror et al., 2017).
- **Interpretability**: While reproducibility depends on the ability to understand and communicate a methodology, interpretability typically refers to the ability to understand the operation or output of a particular instantiation of a model (Wallach, 2016; Lipton, 2016). Interpretability is currently an active area of research in machine learning, and the term is to some extent overloaded (Doshi-Velez and Kim, 2017). However, the key is that interpretability must be evaluated within the context of who is asking and what they want to know. A great deal of work in interpretability focuses on providing approximate explanations to complex models, whereas other seeks to provide true explanations that are nevertheless

simple. Transparency also acts as a foundation for interpretability, as making sense of decisions is much more difficult in the case where the system is completely hidden from view. For additional discussion of interpretability, please refer to Chapter 6.

- **Scale and cost**: Especially in the age of big data, one of the key promises of text-as-data research is the ability to conduct measurements at scale (Salganik, 2017). Most methods that involve some amount of automation will be able to scale to large corpora, whereas those based on human judgments will necessarily be quite limited (though scaling these may be possible through online crowd workers). Closely tied to scale is cost. Almost any method can be made to scale for enough money. Different researchers have different budgets, however, and certain methods may only be feasible for well-funded groups with access to sufficient computational resources. Not only will this determine the types of methods one might consider, it also connects to issues such as reproducibility, as a method which is perfectly reproducible in principle will not be reproducible in practice if the cost is prohibitive. While most traditional social science research has been constrained by relatively small budgets, machine learning research in industry is tending towards ever more computationally intensive models and larger datasets, which has drawn attention to negative externalities (Strubell et al., 2019), and a renewed emphasis on solutions for the limited-resource setting (Gururangan et al., 2019).

As a concrete example to illustrate above desiderata, consider three simple tools for conducting measurements of text: human coding, dictionary methods, and text classification via supervised learning. As mentioned above, human coding (i.e., annotation) is perhaps the most conventional approach to making measurements of text. Because humans are able to easily and effectively process the meaning of text,

they are able to recognize arbitrarily complex categories with some acceptable level of agreement (conditional on appropriate training). They are also able to do some amount of discovery simultaneously, identifying new categories which have not yet been codified, and should be added to the codebook.[4] The codes given by annotators to documents, perhaps combined via some method for aggregating annotations or resolving disputes, thus enact the desired measurements.

Having multiple annotators code each document allows us to quantify the error of this instrument in terms of the rate of agreement, as measured by an appropriate metric, such as Krippendorff's alpha (Krippendorff, 2011). When done properly, this method has great potential to have excellent validity, as humans have the necessary world knowledge and reasoning ability to determine whether or not a piece of text truly expresses a particular concept. However, the limitations of this method are due to the lack of interpretability, reproducibility, and scale. Because annotation is a time-consuming task, in many cases requiring skilled annotators, any annotation effort will be costly and time-consuming, and will thus be limited in its ability to generate more than a relatively small number of measurements. Moreover, no matter how carefully a codebook is constructed, there will always be difficulties in knowing exactly why a particular choice was made, or in trying to apply the same codebook in a new setting. For small projects, human coding is in some sense the gold standard, but comes with severe limitations for more ambitious projects.

Dictionary methods, by contrast, define a set of words associated with each category. These words might be derived from a statistical method, prior intuition, surveys, or any other means (Grimmer and Stewart, 2013). In the simplest setting, we might weight all words equally, and simply count up the number of words from each list that appear in each document. A slightly richer dictionary model would assign a weight to each word.

[4] Note that there is an extensive literature on how to define tasks, train annotators, aggregate judgements, etc. See for example Krippendorff (2012), Barbera et al. (2019), and references therein.

The sum for each document would then be our measurement of that document. This is an incredibly simple approach that is easily understood, reproducible, and scalable.

Dictionary methods are also highly interpretable, in that we can easily look to see which words were responsible for the counts for any document, and they are relatively easy to adapt or extend, by coming up with a new or modified word list appropriate to a new setting. However, the major downsides of this method have to do with its reliability and validity. Because of the simplicity of dictionary methods, they are unlikely to be sufficiently accurate or comprehensive to make correct measurements of all documents. It is still possible that a well-calibrated dictionary method could give an unbiased result in the domain for which it was developed. However, the validity of dictionary methods is open to serious dispute when applied to a new domain (Grimmer and Stewart, 2013). Because a dictionary will have many features that could be specific to the domain for which it was developed (and validated), there is no reason to think that the same dictionary will work well for a new domain.

Finally, consider the use of text classification via supervised learning as a tool for measurement. This idea will be considered in more depth in Chapter 5, but for the moment, consider the conventional use of text classification using a standard machine learning approach, such as logistic regression or deep neural networks. This approach shares some of the advantages of dictionary methods, especially in terms of scalability. Moreover, we might expect that supervised learning would offer greater reliability than a simple dictionary approach, as it could be a more subtle instrument, sensitive to a wider range of evidence. However, depending on the complexity of model used, reproducibility (in the sense of re-creating the same classifier) might be an issue. In some cases, even applying such a model might require specialized hardware such as GPUs. This method may also suffer in terms of interpretability, as the reasons for predictions may be less obvious. Above all, as with dictionaries, we should be concerned

that this method will suffer from domain shift, and problems of validity when applied out of domain. Finally, because we will typically want to train a model for each problem, at least some amount of labeled data will likely be required as an initial step, requiring some of the start-up costs of human annotation.

As evidenced by the above discussion, it should be clear that there is no single method which is most appropriate for all settings, and that we should expect to face a trade off. Researchers will need to choose based on their priorities for each project. As such, the remainder of this thesis is largely about adding nuance to these concerns, and providing specific examples of additional methodologies which expand the menu of options for future work in text-as-data research.

As a final note, it is important to emphasize two additional aspects of computational social science and the text-as-data paradigm. First, although there is a strong tendency to treat text on the internet as found data, we must remember that all such data exists in context and has a history. Naively using such data has the potential to unintentionally recreate *biases* which exist in the data, and which can potentially be amplified by our models (Bolukbasi et al., 2016; Caliskan et al., 2017; Zhao et al., 2017; Sap et al., 2019). Moreover, even if users are aware of website terms and conditions, they may still have additional expectations and preferences about how the data they create and share will be used (Nissenbaum, 2009; Fiesler and Proferes, 2018). We should remember that, as in more traditional research, aggregating data entails potential harm (Ohm, 2010; Salganik, 2017). Identifying patterns may be valuable, but making predictions about individuals may be reckless.

Second, research in the social sciences is noteworthy in that it has the potential to inform policy decisions and lead to real-world impact. Already, we are seeing instances of both research and deployed systems related to criminal sentencing (Kleinberg et al., 2017a), health care (Ustun and Rudin, 2016), and predictive policing (Wang and Rudin,

2015), in many cases using proprietary software. This makes it all the more important that we keep in mind aspects of fairness, accountability, transparency, and ethics of our models (Hardt et al., 2016; Corbett-Davies et al., 2017; Selbst et al., 2019). Although computational social science allows us to ask new questions, or answer the same questions at lower cost (Salganik, 2017), computational methods must be used with care, and results should be treated with skepticism in the absence of validation, as with all research.

2.4 Running example: Framing in the media

As a running case study that I will draw on throughout this thesis, consider the issue of *framing*. A long tradition of research in multiple disciplines has demonstrated that the presentation of information, especially in narrative form, is rarely, if ever neutral (Gitlin, 1980; Entman, 1993; Benford and Snow, 2000; Chong and Druckman, 2007; D'Angelo and Kuypers, 2010). Rather, this use of language inherently involves choices about what to report, how to characterize people and events, what background facts to include, the use of metaphors, and so on. Although there are plenty of examples of people who engage in deliberate framing (e.g., Lakoff et al., 2008) we are generally most interested in more systemic, background effects – persistent patterns in how people communicate about issues that change slowly over time.

In a widely cited definition, Entman (1993) argues that "to frame is to select some aspects of a perceived reality and make them more salient in a communicating text, in such a way as to promote a particular problem definition, causal interpretation, moral evaluation, and/or treatment recommendation." Further elaborations have emphasized how various elements of framing tend to align and cohere, eventually being deployed as "packages" which can be evoked through particular phrases, images,

or other synecdoches (Gameson and Modigliani, 1989; Benford and Snow, 2000; Chong and Druckman, 2007).

Framing is a phenomenon largely studied and debated in the social sciences, where it is common to analyze specific issues in meticulous detail. Past work on framing includes many examples of issue-specific studies based on manual content analysis (Baumgartner et al., 2008; Berinsky and Kinder, 2006). While such studies reveal much about the range of opinions on specific issues, such as the death penalty, they do not characterize framing at a level of abstraction that allows comparison across social issues.

More recently, there has been a surge of interest in framing within the NLP community (Nguyen et al., 2015c; Tsur et al., 2015; Baumer et al., 2015; Field et al., 2018; Demszky et al., 2019; Hartmann et al., 2019), and I have argued elsewhere that framing can be understood as a general aspect of linguistic communication about facts and opinions on any issue (Card et al., 2015). Moreover, it is important because a) it has been repeatedly demonstrated that framing has an impact on people's expressed opinions, at least in the short term (Hopkins and Mummolo, 2017); and b) it is commonly held that the dominant framing in mainstream media tends to reflect the preferences of powerful institutions and actors (Herman and Chomsky, 1988). By making use of larger corpora than could be analyzed by manual close reading, we may be in a position to test and/or add nuance to both of these theories.

As a starting point towards analyzing framing as a general phenomenon that operates across issues, I have created a dataset in collaboration with others that we have called the Media Frames Corpus (MFC), which will be used for experiments in some chapters of this thesis. The MFC collects news articles from major U.S. newspapers on a set of six issues, for each of which we have richly annotated thousands of articles in terms of a set of 15 cross-cutting framing dimensions (Card et al., 2015). These

categories are designed to subsume the more precise issue-specific frames that might be used for any particular issue, and the full list of them is given in Figure 2.2. The annotations are fine-grained, with annotators identifying spans of text which cue particular frames for them, providing a valuable source of information about the instantiation of framing in language. However, there is naturally a degree of subjectivity to this annotation task, and we have preserved disagreements between annotators, which in themselves provide evidence of the diversity of framing effects.

Economic: costs, benefits, or other financial implications

Capacity and resources: availability of physical, human or financial resources, and capacity of current systems

Morality: religious or ethical implications

Fairness and equality: balance or distribution of rights, responsibilities, and resources

Legality, constitutionality and jurisprudence: rights, freedoms, and authority of individuals, corporations, and government

Policy prescription and evaluation: discussion of specific policies aimed at addressing problems

Crime and punishment: effectiveness and implications of laws and their enforcement

Security and defense: threats to welfare of the individual, community, or nation

Health and safety: health care, sanitation, public safety

Quality of life: threats and opportunities for the individual's wealth, happiness, and well-being

Cultural identity: traditions, customs, or values of a social group in relation to a policy issue

Public opinion: attitudes and opinions of the general public, including polling and demographics

Political: considerations related to politics and politicians, including lobbying, elections, and attempts to sway voters

External regulation and reputation: international reputation or foreign policy of the U.S.

Other: any coherent group of frames not covered by the above categories

Figure 2.2: Framing dimensions from Boydstun et al. (2014).

Chapter 3

Inferring character and story types as an aspect of framing

(This chapter was originally published as Card et al., 2016)

3.1 Introduction

As discussed in Chapter 2, communication inescapably involves *framing*—choosing "a few elements of perceived reality and assembling a narrative that highlights connections among them to promote a particular interpretation" (Entman, 2007). Memorable examples include loaded phrases (e.g., "death tax", "war on terror"), but the literature attests a much wider range of linguistic means toward this end (Pan and Kosicki, 1993; Greene and Resnik, 2009; Choi et al., 2012; Baumer et al., 2015).

Framing is associated with several phenomena to which NLP has been applied, including *ideology* (Lin et al., 2006; Hardisty et al., 2010; Iyyer et al., 2014; Preotiuc-Pietro et al., 2017), *sentiment* (Pang and Lee, 2008; Feldman, 2013), and *stance* (Walker et al.,

2012; Hasan and Ng, 2013). Although such author attributes are interesting, framing scholarship is concerned with persistent patterns of representation of particular issues—without necessarily tying these to the states or intentions of authors—and the effects that such patterns may have on public opinion and policy. Note that NLP has also often been used in large-scale studies of news and its relation to other social phenomena (Leskovec et al., 2009; Gentzkow and Shapiro, 2010; Smith et al., 2013; Niculae et al., 2015).

Can framing be automatically recognized? If so, social-scientific studies of framing will be enabled by new *measurements*, and new applications might bring framing effects to the consciousness of everyday readers. Research in NLP has explored unsupervised framing analysis of political text using autoregressive and hierarchical topic models (Nguyen et al., 2013, 2015c; Tsur et al., 2015), but most of these conceptualize framing along a single dimension. Rather than trying to place individual articles on a continuum from liberal to conservative or positive to negative, I am interested in discovering broad-based patterns in the ways in which the media communicate about issues.

In this chapter, my focus is on the narratives found in news stories, specifically the participants in those stories. Insofar as journalists make use of archetypal narratives (e.g., the struggle of an individual against a more powerful adversary), one would expect to see recurring representations of characters in these narratives (Schneider and Ingram, 1993; Van Gorp, 2010). A classic example is the contrast between "worthy" and "unworthy" victims (Herman and Chomsky, 1988). As another example, Glenn Greenwald emphasized how he was repeatedly characterized as an "activist" or "blogger", rather than a "journalist" during his reporting on the NSA (Greenwald, 2014).

The model I present here builds on the "Dirichlet persona model" (DPM) introduced by Bamman et al. (2013) for the unsupervised discovery of what they called *personas* in short film summaries (e.g., the "dark hero"). As in the DPM, I operationalize personas

as mixture of textually-expressed characteristics of entities: what they do, what is done to them, and their descriptive attributes. I begin by providing a description of the full model, after which I highlight the differences from the DPM.

This chapter presents an example of a model designed specifically to enable both a new way of exploring a corpus, and a particular type of measurement from text, in this case one discovered automatically and focused on how entities are represented. The main contributions are:

- I strengthen the DPM's assumptions about the *combinations* of personas found in documents, applying a Dirichlet process prior to infer patterns of coocurrence (§3.3). The result is a clustering of documents based on the collections of personas they use, discovered simultaneously with those personas.
- Going beyond named characters, I allow Bamman-style personas to account for entities like institutions, objects, and concepts (§3.5).
- I demonstrate that this model produces *interpretable* clusters which provide insight into the immigration articles in the Media Frames Corpus (MFC; §3.6).
- I propose a new kind of evaluation based on Bayesian optimization. Given a supervised learning problem, I treat the inclusion of a candidate feature set (here, personas) as a hyperparameter to be optimized alongside other hyperparameters (§3.7).
- In the case of U.S. news stories about immigration, I find that personas are, in many cases, helpful for automatically inferring the coarse-grained framing and tone employed in a piece of text, as defined in the MFC (§3.7). Demonstrating that discovered personas are predictive of human-annotated frames and tone can be seen as a type of validation of the proposed form of measurement.

3.2 Model description

The plate diagram for the model presented in this chapter is shown in Figure 3.1 (right), with the original DPM (Bamman et al., 2013) shown on the left.

As evidence, the model considers tuples $\langle w, r, e, i \rangle$, where w is a word token, with r being the category of syntactic relation[1] it bears to an entity with index e mentioned in document with index i. The model's generative story explains this evidence as follows:

1. Let there be K topics as in LDA (Blei et al., 2003). Each topic $\phi_k \sim \mathrm{Dir}(\gamma)$ is a multinomial over the V words in the vocabulary, drawn from a Dirichlet parameterized by γ.
2. For each of P personas p, and for each syntactic relation type r, define a multinomial $\psi_{p,r} \sim \mathrm{Dir}(\beta)$ over the K topics, each drawn from a Dirichlet parameterized by β.
3. Assume an infinite set of distributions over personas drawn from a base distribution H. Each of these $\theta_j \sim \mathrm{Dir}(\alpha)$ is a multinomial over the P personas, with an associated probability of being selected π_j, drawn from the stick-breaking process with hyperparameter λ.
4. For each document i:
   (a) Draw a cluster assignment $s_i \sim \pi$, with corresponding multinomial distribution over personas θ_{s_i}.
   (b) For each entity e participating in i:
      i. Draw e's persona $p_e \sim \theta_{s_i}$.
      ii. For every $\langle r, w \rangle$ tuple associated with e in i, draw $z \sim \psi_{p_e,r}$ then $w \sim \phi_z$.

[1] I adopt the terminology from Bamman et al. (2013) of "agent", "patient", and "attribute", even though these categories of relations are defined in terms of syntactic dependences.

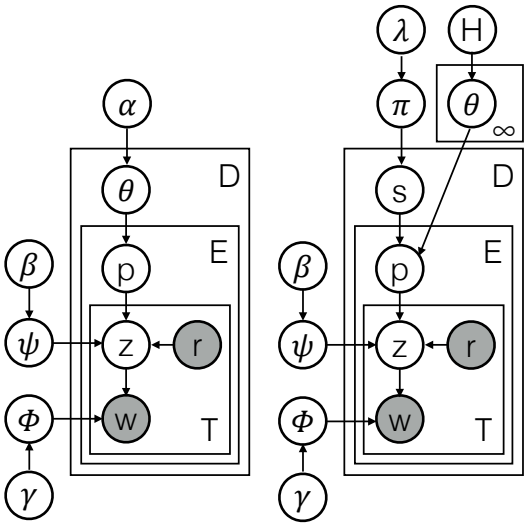


Figure 3.1: Plate diagrams for the DPM (left), and for the new model (right).

The DPM (Figure 3.1, left) has a similar generative story, except that each document has a unique distribution over personas. As such, step 4(a) is replaced with a draw from a symmetric Dirichlet distribution $\theta_i \sim \text{Dir}(\alpha)$.

3.3 Clustering stories

The DPM assumes that each document has a unique distribution (θ_i) from which its personas are drawn. However, for entities mentioned in news articles (as well as for the *dramatis personae* of films), one would expect certain types of personas to occur together frequently, such as articles about lawmakers and laws. Thus I would like to cluster documents based on their "casts" of personas. To do this, I have added a Dirichlet process (DP) prior on the document-specific distribution over personas (step 3), which allows the number of clusters to adapt to the size and complexity of the corpus (Antoniak, 1974; Escobar and West, 1994).

There are a number of equivalent formulations of Dirichlet process mixture models.

Here I present the formulation based on the stick-breaking process. According to this perspective, each mixture component is drawn from an (infinite) set of mixture components (equivalently, clusters), each of which is drawn from a base measure, H. The conditional probability of a cluster assignment is distributed according to an (infinite) multinomial distribution, generated according to the stick-breaking process, with hyperparameter λ. In particular,

$$\{\pi'_k\}_{k=1}^{\infty} \sim \text{Beta}(1, \lambda) \tag{3.1}$$

$$\{\pi_k\}_{k=1}^{\infty} \sim \pi'_k \prod_{l=1}^{k-1}(1 - \pi'_l) \tag{3.2}$$

$$s_i \sim \boldsymbol{\pi} \tag{3.3}$$

$$\theta_{s_i} \sim H \tag{3.4}$$

In this model, I take H to be a symmetric Dirichlet distribution with hyperparameter α. Given a cluster assignment for the ith document, each entity's persona is then drawn according to $p_e \sim \theta_{s_i}$, where s_i indexes the cluster assignment of the ith document.

Although the model admits an unbounded number of distributions over personas, the properties of DPs are such that the number used by D documents will tend to be much less than D. As a result, inference under this model provides topics ϕ (distributions over words) interpretable as textual descriptors of entities, personas ψ (distributions over reusable topics), and clusters of articles s with associated distributions over personas θ.

Following Bamman et al. (2013), I perform inference using collapsed Gibbs sampling, collapsing out the distributions over words (ϕ), topics (ψ), and personas (θ), as well as π. On each iteration, I first sample a cluster for each document, followed by a persona for each entity, followed by a topic for each tuple. Because I assume a conjugate

base measure, sampling clusters can be done efficiently using the Chinese restaurant process (Aldous, 1985) for story types, personas, and topics, with slice sampling for hyperparameters (α, β, γ, λ).

The probability of a document being assigned to an existing cluster (mixture component) is proportional to the number of documents already assigned to that cluster times the likelihood of the document's current personas being generated from that cluster's distribution over personas (θ_{s_i}). The probability of the document being assigned to a new cluster is proportional to λ times the likelihood of the document's personas being generated from a new draw from the base distribution. Integrating out θ and π gives:

$$p(s_i = s' \mid \mathbf{s}_{-i}, \mathbf{p}, \alpha, \lambda) \propto n_{s',*}^{(-i)} \times f(s') \tag{3.5}$$

$$p(s_i = s^{new} \mid \mathbf{s}_{-i}, \mathbf{p}, \alpha, \lambda) \propto \lambda \times f(s^{new}) \tag{3.6}$$

$$f(s) = \prod_{j=1}^{J} \frac{\alpha + n_{s,p_j}^{(-i)} + \sum_{j'=1}^{j-1} \mathbb{I}[p'_j = p_j]}{P\alpha + n_{s,*}^{(-i)} + (j-1)} \tag{3.7}$$

Here, s' is an existing cluster, J ranges over the entities in document i, and p_j is the persona of the jth entity. $n_{s,p_j}^{(-i)}$ is the number of entities in documents of type s with persona p_j, excluding those in i. $n_{s,*}^{(-i)}$ is the total number of entities in this set, and $\mathbb{I}[\cdot]$ is the indicator function.

The equation for sampling personas is similar, and can be shown to be

$$p(p_e = p \mid \mathbf{p}_{-e}, \mathbf{z}, s_e, \alpha, \beta) = \left(\alpha + n_{s_e,p}^{(-e)}\right) \times \prod_{r=1}^{R} \prod_{t=1}^{T_{e,r}} \frac{\beta + n_{p,r,k_t}^{(-e)} + \sum_{t'=1}^{t-1} \mathbb{I}[k_{t'} = k_t]}{K\beta + n_{p,r,*}^{(-e)} + (t-1)} \tag{3.8}$$

where $n_{s_e,p}^{(-e)}$ is the number of entities with persona p in documents with cluster s_e, excluding entity e. R is the number of categories of relations (agent, patient, attribute), $T_{e,r}$ is the number of tuples for entity e with relation r, k_t is the topic of the tth tuple in $T_{e,r}$, $n_{p,r,k_t}^{(-e)}$ is the number of tuples with relation r and topic k_t for entities with persona

p, excluding entity e, and $n_{p,r,*}^{(-e)}$ are these counts summed over topics.

The equation for sampling the topic of a tuple attached to entity e is:

$$p(z_t = k \mid \mathbf{z}_{-t}, \mathbf{p}, \mathbf{w}, \mathbf{r}, \beta, \gamma) = \left(\beta + n_{p_e,r_t,k}^{(-t)}\right) \times \frac{\gamma + n_{k,w_t}^{(-t)}}{V\gamma + n_{k,*}^{(-t)}} \tag{3.9}$$

where V is the size of the vocabulary, $n_{p_t,r_t,k}^{(-t)}$ is the number of tuples with relation r_t and topic k attached to entities with persona p_e, excluding tuple t. $n_{k,w_t}^{(-t)}$ is the number of tuples with w_t assigned to topic k, excluding t, and $n_{k,*}^{(-t)}$ is the sum of these counts across topics.

During sampling, I discard samples from the first 10,000 iterations, and collect one sample from every tenth iteration for following 1,000 iterations. I sample hyperparameters every 20 iterations for the first 500 iterations, and every 100 thereafter.

3.4 Dataset

For the experiments this chapter, I use the articles about immigration from the MFC, as described in in §2.4. Specifically, I make use of the annotations of the "primary frame" (one of 15 general-purpose "framing dimensions", such as *Politics* or *Legality*) and "tone" (*pro*, *neutral*, or *anti*) of each article for a set of approximately 4,200 annotated articles. In order to train this model on a larger collection of articles, I use the original corpus of articles from which the annotated articles in the MFC were drawn. This produces a corpus of approximately 37,000 articles about immigration; I train the persona model on this larger dataset, only using the smaller set for evaluation on a secondary task. Note that the MFC annotations are not used by this model; rather, I hypothesize that the personas it discovers may serve as features to help predict framing—this serves as one of my evaluations (§3.7).

3.5 Identifying entities

The original focus of the DPM was on *named* characters in movies, which could be identified using named entity recognition and pronominal coreference (Bamman et al., 2013), or name matching for pre-defined characters (Bamman et al., 2014). Here, I are interested in applying this model to entities about which I assume no specific prior knowledge.

In order to include a broader set of entities, I preprocess the corpus and apply a series of filters. First, I obtain lemmas, part-of-speech tags, dependencies, coreference resolution, and named entities from the Stanford CoreNLP pipeline (Manning et al., 2014), as well as supersense tags from the AMALGrAM tagger (Schneider and Smith, 2015). For each document, I consider all tokens with a NN* or PRP part of speech as possible entities, partially clustered by coreference. I then merge all clusters (including singletons) within each document that share a non-pronomial mention word.

Next, I exclude all clusters lacking at least one mention classified as a person, organization, location, group, object, artifact, process, or act (by CoreNLP or AMALGrAM). From these, I extract $\langle w, r, e, i \rangle$ tuples using extraction patterns lightly adapted from (Bamman et al., 2013). The complete set of patterns I use are given in the Table 3.1. To further restrict the set of entities to those that have sufficient evidence, I construct a vocabulary for each of the three relations, and exclude words that appear less than three times in the corresponding vocabulary.[2] I then apply one last filter to exclude entities that have fewer than three qualifying tuples across all mentions. From the dataset described in §3.4, I extract 128,655 entities, mentioned using 11,262 different mention words, with 575,910 tuples and 11,104 distinct $\langle r, w \rangle$ pairs.

[2] I also exclude the lemma "say" as a stopword, as it is the most common verb in the corpus by an order of magnitude

Relation type	Neighbor	Arc type	POS
Attribute	parent	nsubj	JJ
	parent	nsubj	NN*
	child	amod	JJ
Agent	parent	agent	VB*
	parent	nsubj	VB*
	child	acl	VB*
Patient	parent	dobj	VB*
	parent	nsubjpass	VB*
	parent	iobj	VB*

Table 3.1: Extraction patterns.

3.6 Exploratory analysis

Here I discuss the model presented in this chapter, as estimated on the corpus of 37,000 articles discussed in §3.4 with 50 personas and 100 topics; these values were not tuned. A cursory examination of topics shows that each tends to be a group of either verbs or attributes. Personas, on the other hand, blend topics to include all three relation types. The estimated Dirichlet hyperparameters are all $\ll 1$, giving sparse (and hence easily scanned) distributions over personas, topics, and words.

Table 3.2 shows all 50 personas. For each p, I show (i) the mention words most strongly associated with p, and (ii) $\langle r, w \rangle$ pairs associated with the persona. (To save space, "I" denotes *immigrant*.) Recall that, like the Dirichlet persona model, my model says nothing about the mention words; they are *not* included as evidence during inference.[3] Nonetheless, each persona is strongly associated with a sparse handful of mention words, and I find that labeling each persona by its most strongly associated mention word (excluding *immigrant*) is often sensible (these are capitalized in Table 3.2, though in some cases the relation words differentiate strongly (e.g., the *group* personas, IDs 17 and 18 in Table 3.2).

[3] I did explore adding mention words as evidence, but they tended to dominate the relation tuples. Because my interest is in a richer set of framing devices than simply the words used to refer to people (and other entities), I consider here only the model based on the surrounding context.

ID	Mention words	Relations
1	AGENT police official authority	federal$_m$ tell$_p$ find$_a$ arrest$_a$ local$_m$ tell$_a$
2	ASYLUM crime refugee asylum_seeker	political$_m$ seek$_p$ grant$_p$ commit$_p$ serious$_m$ deny$_p$
3	BILL law immigration_reform measure	comprehensive$_m$ pass$_a$ pass$_p$ make$_a$ have$_a$ support$_p$
4	BOAT van crime document	criminal$_m$ other$_m$ have$_p$ use$_a$ use$_p$ be$_a$
5	BORDER border_patrol border_agent	mexican$_m$ cross$_p$ secure$_p$ southern$_m$ u.s.-mexico$_m$
6	BUSH official mcnary people I	have$_a$ tell$_a$ want$_a$ tell$_p$ former$_m$ call$_a$
7	CANDIDATE bush romney leader	republican$_m$ presidential$_m$ democratic$_m$ have$_a$ call$_a$
8	CARD document visa status	green$_m$ new$_m$ get$_p$ temporary$_m$ fake$_m$ permanent$_m$
9	CARD visa state document	consular$_m$ federal$_m$ have$_a$ mexican$_m$ receive$_p$ get$_p$
10	COMPANY country I state nation	have$_a$ regional$_m$ global$_m$ rural$_m$ take$_a$ require$_p$
11	COUNTRY people I citizen united_states	american$_m$ other$_m$ enter$_p$ have$_a$ leave$_p$ central$_m$
12	COUPLE marriage people I class	gay$_m$ bilingual$_m$ same-sex$_m$ have$_a$ prime$_m$
13	COURT lawsuit suit ruling	federal$_m$ file$_p$ rule$_a$ civil$_m$ file$_a$ have$_a$
14	EMPLOYER company people business	hire$_a$ have$_a$ many$_m$ require$_p$ employ$_a$ local$_m$
15	FENCE amendment law wall	real$_m$ 14th$_m$ virtual$_m$ build$_p$ be$_a$ have$_a$
16	GOVERNMENT court judge official	federal$_m$ local$_m$ have$_a$ rule$_a$ ask$_p$ other$_m$
17	GROUP deportation attack country	terrorist$_m$ civil$_m$ face$_p$ armed$_m$ islamic$_m$ muslim$_m$
18	GROUP I voter people bush	hispanic$_m$ immigrant$_m$ local$_m$ many$_m$ want$_a$ have$_a$
19	I ALIEN immigration people worker	illegal$_m$ allow$_p$ have$_a$ legal$_m$ undocumented$_m$ live$_a$
20	I ALIEN people criminal inmate	illegal$_m$ criminal$_m$ deport$_p$ immigrant$_m$ detain$_p$
21	I ALIEN worker immigration employer	illegal$_m$ hire$_p$ undocumented$_m$ employ$_p$ legal$_m$
22	I ALIEN worker people immigration	illegal$_m$ arrest$_p$ undocumented$_m$ arrest$_a$ charge$_p$
23	I CHILD worker people student	immigrant$_m$ foreign-born$_m$ have$_a$ many$_m$ come$_a$
24	I GROUP people population business	new$_m$ immigrant$_m$ other$_m$ many$_m$ asian$_m$ have$_a$
25	I GROUP program center city	new$_m$ have$_a$ first$_m$ be$_a$ other$_m$ make$_a$
26	I IMMIGRATION alien worker	illegal$_m$ legal$_m$ hire$_p$ have$_a$ allow$_p$ undocumented$_m$
27	I IMMIGRATION alien worker people	illegal$_m$ legal$_m$ have$_a$ be$_a$ come$_a$ immigrant$_m$
28	I JEWS refugee israel child	soviet$_m$ jewish$_m$ russian$_m$ have$_a$ vietnamese$_m$
29	I MAN alien refugee people	illegal$_m$ chinese$_m$ cuban$_m$ arrest$_p$ haitian$_m$ find$_p$
30	I PEOPLE child student worker	many$_m$ young$_m$ have$_a$ illegal$_m$ come$_a$ be$_a$
31	I PEOPLE country woman man	black$_m$ muslim$_m$ african$_m$ have$_a$ come$_a$ korean$_m$
32	I WORKER people citizen job	american$_m$ new$_m$ have$_a$ mexican$_m$ illegal$_m$ many$_m$
33	I WORKER resident student people	legal$_m$ foreign$_m$ permanent$_m$ have$_a$ allow$_p$ skilled$_m$
34	I WORKER student people child	undocumented$_m$ illegal$_m$ immigrant$_m$ have$_a$ allow$_p$
35	JOB I people immigration law	have$_p$ have$_a$ be$_a$ take$_p$ good$_m$ make$_a$
36	JOB study survey I labor	find$_a$ new$_m$ find$_p$ show$_a$ fill$_p$ take$_p$
37	LAW immigration_law bill measure	new$_m$ federal$_m$ enforce$_p$ require$_a$ pass$_p$ allow$_a$
38	MAN I woman people haitians	deport$_p$ have$_a$ arrest$_p$ hold$_p$ release$_p$ face$_a$
39	MAN people agent official I	arrest$_p$ charge$_p$ other$_m$ former$_m$ have$_a$ face$_a$
40	MAN woman I people girl	tell$_a$ kill$_p$ have$_a$ other$_m$ young$_m$ take$_p$
41	PEOPLE I child man woman	have$_a$ come$_a$ live$_a$ go$_a$ tell$_p$ work$_a$
42	PROFILING violence abuse discrimination	racial$_m$ domestic$_m$ safe$_m$ physical$_m$ be$_a$ affordable$_m$
43	PROGRAM system law agency	new$_m$ national$_m$ federal$_m$ create$_p$ use$_p$ special$_m$
44	REFUGEE I boy people elian	cuban$_m$ haitian$_m$ chinese$_m$ have$_a$ allow$_p$ return$_p$
45	SCHOOL people I family english	have$_a$ high$_m$ see$_a$ come$_a$ go$_a$ be$_a$
46	SERVICE school care college	public$_m$ medical$_m$ provide$_p$ deny$_p$ receive$_p$ attend$_p$
47	TRAFFICKING rights group flight	human$_m$ international$_m$ commercial$_m$ be$_a$ have$_a$
48	WORKER I immigration student company	foreign$_m$ legal$_m$ skilled$_m$ hire$_p$ american$_m$ have$_a$
49	WORKER I people woman man	mexican$_m$ immigrant$_m$ undocumented$_m$ migrant$_m$
50	YEAR program month income	fiscal$_m$ last$_m$ end$_a$ next$_m$ previous$_m$ begin$_a$

Table 3.2: Personas with their associated mention words and relation tuples (a = agent, p patient, m = modifier/attribute); I denotes "immigrant".

In general, the output of the model appears to have reasonably strong face validity. For example, the model finds expected participants (such as *workers*, political *candidates*, and *refugees*), but also more conceptual entities, such as *laws*, *bills* (IDs 3, 37), and the U.S.-Mexican *border* (ID 5), which looms large in the immigration debate. Some interesting distinctions are discovered, such as two of the *worker* personas, one high-skilled and residing legally (ID 48), the other illegal (ID 49).

Using the original publication dates of the articles, I can estimate the frequency of appearance of each persona within immigration coverage by summing the posterior distribution over personas for each entity mention, and plotting these frequencies across time. (Note that time metadata is not given to the model as evidence.) I find immediately that personas can signal events. Figure 3.2 shows these temporal trajectories for a small, selected set of personas. Although *bills* and *laws* are conceptually similar, and have similar trajectories from 1980 to 2005, they are strongly divergent in 2006 and 2010. These are particularly notable years for immigration policy, corresponding to the failed Comprehensive Immigration Reform Act of 2006 (Senate bill S.2611) and Arizona's controversial anti-immigration laws from 2010.[4] Refugees, by contrast, show a marked spike around the year 2000. Inspection showed this persona to be strongly tied to the case of Elián González, which received a great deal of media attention in that year.

The main advantage of the extended model over the DPM is being able to cluster articles by "casts". During sampling, thousands of clusters are created (and mostly destroyed). Ultimately, this inference procedure settled on approximately 110 clusters, and I consider two examples. Figure 3.3 shows the temporal trajectories of the two clusters with the greatest representation of the *refugee* persona. Both show the characteristic spike around the year 2000. The top personas for these two clusters are given

[4] Other notable events which appear to be represented include the Illegal Immigration Reform and Immigrant Responsibility Act of 1996, and the Secure Fence Act of 2006.

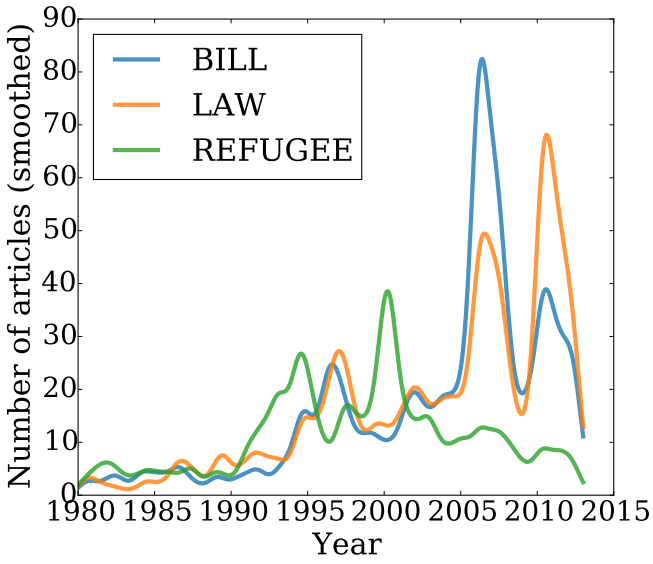


Figure 3.2: Temporal patterns of the mentions of selected personas.

in Table 3.3. Type A, which includes a story with the headline "Protesters vow to keep Elián in U.S.," emphasizes political aspects, while type B (e.g., "Court says no to rights for refugees") emphasizes legal aspects. Note that *Political* and *Legality* are two of the framing dimensions used in the MFC.

Do these persona-cast clusters relate to frames? For the five most common story clusters, (which have no overlap with the two refugee story types), Figure 3.4 shows the number of annotated articles with each of the primary frames if I assign each article to its most likely cluster. The second and fifth clusters correlate particularly well with primary frames (*Political* and *Crime*, respectively). This is further reinforced by looking at the most frequent persona for each of these story clusters which are *candidate* (ID 7) for the second and *immigrant* (ID 22), characterized by illegal_m and arrest_p, for the fifth.

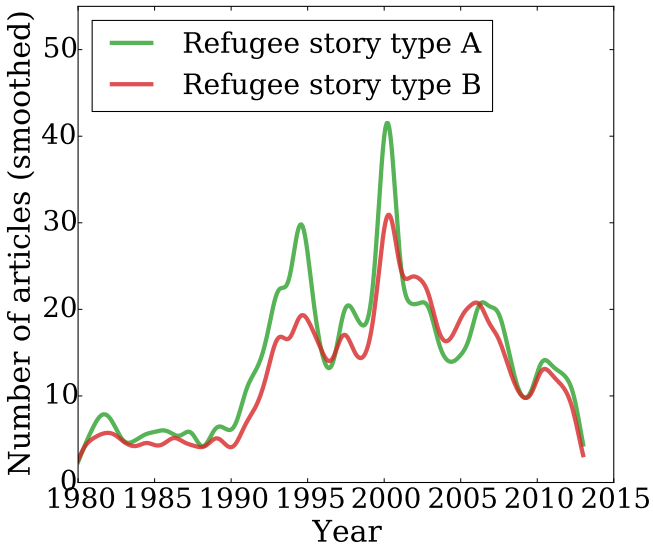


Figure 3.3: Temporal patterns of two clusters with the greatest overall representation of the *refugee* persona.

Refugee story cluster A		
Frequency	Persona	ID
0.49	REFUGEE immigrant boy	44
0.10	BUSH official mcnary	6
0.06	IMMIGRANT man alien	29
0.05	ASYLUM crime refugee	2

Refugee story cluster B		
Frequency	Persona	ID
0.29	MAN immigrant woman	38
0.23	REFUGEE immigrant boy	44
0.12	COURT lawsuit suit	13
0.10	GOVERNMENT court judge	16

Table 3.3: Truncated distribution over personas for the two clusters depicted in Figure 3.3. IDs index into Table 3.2.

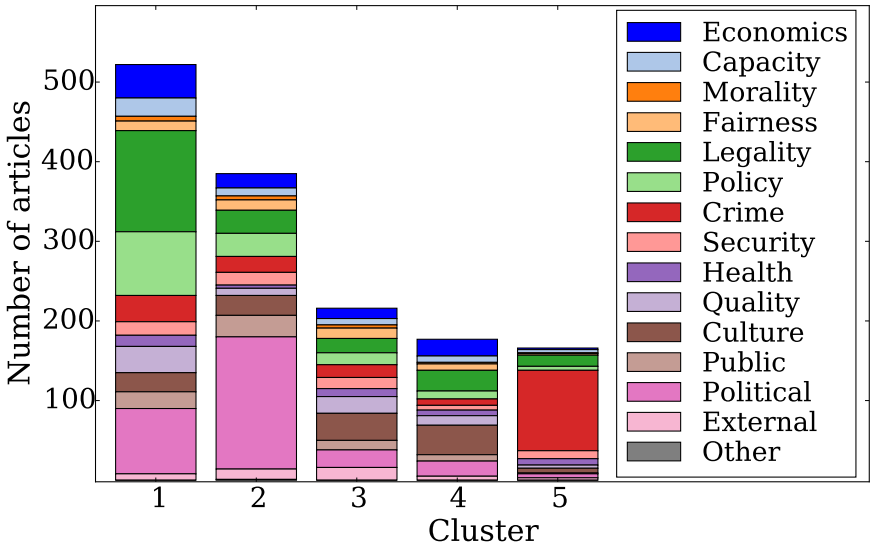


Figure 3.4: Number of annotated articles in each of the five most frequent clusters, with colors showing the proportion of articles annotated with each primary frame.

3.7 Experiments: Personas and framing

For a more quantitative analysis, I evaluate personas as features for automatic analysis of framing and tone, as defined in the MFC (§3.4). Specifically, I build multi-class text classifiers (separately) for the primary frame and the tone of a news article, for which there are 15 and 3 classes, respectively. Because there are only a few thousand annotated articles, I applied 10-fold cross-validation to estimate performance.

Features are derived from this model by considering each persona and each story cluster as a potential feature. A document's feature values for story types are the proportion of samples in which it was assigned to each cluster. Persona feature values are similarly derived by the proportion of samples in which each entity was assigned to each persona, with the persona values for each entity in each document summed into a single set of persona values per document. I did not use the topics (z) discovered by this model as features.

Primary frame			*Tone*		
Feature set	Accuracy	# Features	Feature set	Accuracy	# Features
MF	0.174	0	MF	0.497	0
(W)	0.529	3.9k	(W)	0.628	5.0k
(W, P_1)	0.537	3.5k	(W, P_1)	0.631	5.0k
(W, P_2)	*0.540	3.5k	(W, P_2)	0.628	5.0k
(W, P_2, S)	0.537	2.8k	(W, P_2, S)	0.630	4.0k

Table 3.4: Evaluation using a direct comparison to a simple baseline. Each model uses the union of listed features. (W = unigrams and bigrams, P_1 = personas from DPM, P_2 = personas from my model, S = story clusters; MF = always predict most frequent class.) An asterisk (*) indicates a statistically significant difference compared to the (W) baseline ($p < 0.05$).

3.7.1 Experiment 1: Direct comparison

For the first experiment, I trained independent multi-class logistic regression classifiers for predicting primary frame and tone. I considered adding persona and/or story cluster features to baseline classifiers based only on unigrams and bigrams with binarized counts, a simple but robust baseline (Wang and Manning, 2012).[5] In all cases, I used L_1 regularization and use 5-fold cross validation within each split's training set to determine the strength of regularization. I then repeated this for each of the 10 folds, thereby producing one prediction (of primary frame and tone) for every annotated article. The results of this experiment are given in Table 3.4; for predicting the primary frame, classifiers that used persona and/or story cluster features achieve higher accuracy than the bag-of-words baseline (W); the classifier using personas from my model but not story clusters is significantly better than the baseline.[6] The enhanced models are also more compact, on average, using fewer effective features. A benefit to predicting tone is also observed, but it did not reach statistical significance.

[5] I also binarized the persona feature values.
[6] Two-tailed McNemar's test ($p < 0.05$).

3.7.2 Experiment 2: Automatic evaluation

Although bag-of-n-grams models is a relatively strong simple baseline for text classification, researchers familiar with the extensive catalogue of features offered by NLP will potentially see them as a straw man. I propose a new and more rigorous method of comparison, in which a wide range of features are offered to an automatic model selection algorithm for each of the prediction tasks, with the features to be evaluated withheld from the baseline.

Because no single combination of features and regularization strength is best for all situations, it is an empirical question which features are best for each task. I therefore make use of Bayesian optimization (Bayesopt) to make as many modeling decisions as possible (Pelikan, 2005; Snoek et al., 2012; Bergstra et al., 2015; Yogatama et al., 2015).

In particular, let F be the set of features that might be used as input to any text classification algorithm. Let f be a new feature that is being proposed. Allow the inclusion or exclusion of each feature in the feature set to be a hyperparameter to be optimized, along with any additional decisions such as input transformations (e.g., lowercasing), and feature transformations (e.g., normalization). Using an automatic model selection algorithm such as Bayesian optimization, allow the performance on the validation set to guide choices about all of these hyperparameters on each iteration, and set up two independent experiments.

For the first condition, A_1, allow the algorithm access to all features in F. For the second, A_2, allow the algorithm access to all features in $F \cup f$. After R iterations of each, choose the best model or the best set of models from each of A_1 and A_2 (M_1 and M_2, respectively), based on performance on the validation set. Finally, compare the selected models in terms of performance on the test set (using an appropriate metric such as F_1), and examine the features included in each of the best models. If f is a

helpful feature, one would expect to see that, a) $F_1(M_2) > F_1(M_1)$, and b), f is included in the best model(s) found by A_2.

If $F_1(M_2) > F_1(M_1)$ but f is not included in the best models from A_2, this suggests that the performance improvement may simply be a matter of chance, and there is no evidence that f is helpful. By contrast, if f is included in the best models, but $F_1(M_2)$ is not significantly better than $F_1(M_1)$, this suggests that f is offering some value, perhaps in a more compressed form of the useful signal from other features, but does not actually offer better performance.

For this experiment, I used the tree-structured Parzen estimator for Bayesian optimization (Bergstra et al., 2015), with L_1-regularized logistic regression as the underlying classifier, and set $R = 40$. In addition to the entities and story clusters identified by these models, I allowed these classifiers access to a large set of features, including unigrams, bigrams, parts of speech, named entities, dependency tuples, ordinal sentiment values (Manning et al., 2014), multi-word expressions (Justeson and Katz, 1995), supersense tags (Schneider and Smith, 2015), Brown clusters (Brown et al., 1992), frame semantic features (Das et al., 2010), and topics produced by standard LDA (Blei et al., 2003). The inclusion or exclusion of each feature is determined automatically on each iteration, along with feature transformations (removal of rare words, lowercasing, and binary or normalized counts).

The baseline, denoted "B," offers all features except personas and story clusters to Bayesopt; I consider adding DPM personas, my model's personas, and my model's personas and story clusters. Table 3.5 shows test-set accuracy for each setup, averaged across the three best models returned by Bayesopt.

Using this more rigorous form of evaluation, approximately the same accuracy is obtained in all experimental conditions. However, we can still gain insight into which features are useful by examining those selected by the best models in each condition.

Feature set	Primary Frame	Tone
(B)	0.566	0.667
(B, P_1)	0.568	0.671
(B, P_2)	0.568	0.667
(B, P_2, S)	0.567	0.671

Table 3.5: Mean accuracy of the best three iterations from Bayesian optimization (chosen based on validation accuracy). (B = features from many NLP tools, P_1=personas from the DPM, P_2 = personas from my model, S=story clusters.)

For primary frame prediction, both personas and story clusters are included by the best models in every case where they have been offered as possible features, as are unigrams, dependency tuples, and semantic frames. Other commonly-selected features include bigrams and part of speech tags. For predicting tone, personas are only included by half of the best models, with the most common features being unigrams, bigrams, semantic frames, and Brown clusters. As expected, the best models in each condition obtain better performance than the models from experiment 1, thanks to the inclusion of additional features and transformations.

This secondary evaluation suggests that for this task, persona features are useful in predicting the primary frame, but are unable to offer improved performance over existing features, such as semantic frames. However, the fact that that both personas and story clusters are included by all the best models for predicting the primary frame suggests that they are competitive with other features, and perhaps offer useful information in a more compact form.

Here, Bayesopt has provided a means of evaluating the utility of different *features*. However, it could in principle be used to evaluate the impact of any hyperparameter, such as the the use of dropout in training neural networks. By using the same procedure, and comparing across many datasets, one could establish whether a particular choice is better in expectation. Note however, that one must be careful about comparing models from a finite number of trials. An expanded hyperparameter search space entails the

possibility of a better optimum, but may also require more trials of hyperparameter tuning in order to find a superior result. Thus, we should recognize that performance is a function of the amount of effort used to tune hyperparameters and provide details about this where appropriate (Dodge et al., 2019).

3.8 Qualitative evaluation

Prior to exposure to any output of my model, a contributor to this work (Justin Gross, who has expertise in both framing and the immigration issue) prepared a list of personas he expected to frequently occur in American news coverage of immigration. Given the example of the "skilled immigrant," he listed 22 additional named personas, along with a few examples of things they do, things done to them, and attributes.

The list he prepared includes several different characterizations of immigrants (low-skilled, unauthorized, legal, citizen children, undocumented children, refugees, naturalized citizens), non-immigrant personas (U.S. workers, smugglers, politicians, officials, border patrol, vigilantes), related pairs (pro / anti advocacy groups, employers / guest workers, criminals / victims), and a few more conceptual entities (the border, bills, executive actions). Of these, almost all are arguably represented in the personas that were discovered. However, there is rarely a perfect one-to-one mapping: predefined personas are sometimes merged (e.g., "the border" and "border patrols") or split (e.g., legislation, employers, and various categories of immigrants). Personas which don't emerge from my model include smugglers, guest workers, vigilantes, and victims of immigrant criminals. On the other hand, my model proposes far more non-person entities, such as ID cards, courts, companies, jobs, and programs.

These partial matchings between predefined personas and the results of my model are generally identifiable by comparing the names given to the predefined personas

to the the most commonly occurring mention words and attributes of the discovered personas. The attributes and action words given to the predefined personas are harder to evaluate, as many of them are rare (e.g. politicians "vacillate") or compound phrases (e.g. low-skilled immigrants "do jobs Americans won't do") that tend to miss the more obvious properties captured by my model. For example, the *employer* persona captured by my model engages in actions like *hire*, *employ*, and *pay*. By contrast, the terms given for the pre-defined "business owners" persona are "lobby" and "rely on immigrant labor." The unsupervised discovery of this persona can clearly be matched to the predefined persona in this case, but doesn't provide such fine-grained insight into how they might be characterized.

The best match between predefined and discovered personas is the U.S.-Mexican border. Of the words given for the predefined persona, almost all are more frequently associated with *border* than with any other discovered persona ("Mexican-U.S.," "lawless," "porous," "unprotected," "guarded," and "militarized"). The most commonly associated words discovered by my model that are missing from the predefined description include *crossed*, *secured*, *southern*, and *closed*.

While this qualitative evaluation helps to demonstrate the face validity of my model, it would be better to have a more comprehensive set of predefined personas, based on input from additional experts. Moreover, it also illustrates the challenge of trying to match the output of an unsupervised model to expected results. Not only is some merging and splitting of categories inevitable, there was a mismatch in this case in the types of entities to be described (people as opposed to more abstract entities), and the ways of describing them (rare but specific words as opposed to more generic but potentially obvious terms).

3.9 Related work

Much NLP has focused on identifying entities or events (Ratinov and Roth, 2009; Ritter et al., 2012), analyzing schemes or narrative events in terms of characters (Chambers and Jurafsky, 2009), inferring the relationships between entities (O'Connor et al., 2013; Iyyer et al., 2016), and predicting personality types from text (Flekova and Gurevych, 2015). Bamman also applied variants of the DPM to characters in novels (Bamman et al., 2014), and released a dataset of annotated entities in fiction (Bamman et al., 2019).

Previous work on sentiment, stance, and opinion mining has focused on recognizing stance or political sentiment in online ideological debates (Somasundaran and Wiebe, 2010; Hasan and Ng, 2014; Sridhar et al., 2015), and other forms of social media (O'Connor et al., 2010; Agarwal et al., 2011), and recently through the lens of connotation frames (Rashkin et al., 2016). Opinion mining and sentiment analysis are the subject of ongoing research in NLP and have long served as test platforms for new methodologies (Socher et al., 2013; İrsoy and Cardie, 2014; Tai et al., 2015)

3.10 Summary

In this chapter, I have presented an extension of a model for discovering latent personas to simultaneously cluster documents by their "casts" of personas. By exploring the model's inferences and by incorporating the learned representations as features into a challenging text analysis task—characterizing coarse-grained framing in news articles—I have demonstrated that personas are a useful abstraction when applying NLP to social-scientific inquiry. Finally, I introduced a Bayesian optimization approach to rigorously assess the usefulness of new features in machine learning tasks.

Chapter 4

Modeling documents with metadata using neural variational inference

(This chapter was originally published as Card et al., 2018)

4.1 Introduction

Topic models comprise a family of methods for uncovering latent structure in text corpora, and are widely used tools in the digital humanities, political science, and other related fields (Boyd-Graber et al., 2017), both for topic discovery, and as a way of making measurements of text (O'Connor et al., 2011; Wallach, 2016). Latent Dirichlet allocation (LDA; Blei et al., 2003) is often used when there is no prior knowledge about a corpus. In the real world, however, most documents have non-textual attributes such as author (Rosen-Zvi et al., 2004), timestamp (Blei and Lafferty, 2006), rating (McAuliffe and Blei, 2008), or ideology (Eisenstein et al., 2011; Nguyen et al., 2015c), which I refer to as *metadata*.

Many customizations of LDA have been developed to incorporate document metadata. Two models of note are supervised LDA (SLDA; McAuliffe and Blei, 2008), which jointly models words and labels (e.g., ratings) as being generated from a latent representation, and sparse additive generative models (SAGE; Eisenstein et al., 2011), which assumes that observed covariates (e.g., author ideology) have a sparse effect on the relative probabilities of words given topics. The structural topic model (STM; Roberts et al., 2014), which adds correlations between topics to SAGE, is also widely used, but like SAGE it is limited in the types of metadata it can efficiently make use of, and how that metadata is used. Note that in this work I will distinguish *labels* (metadata that are generated jointly with words from latent topic representations) from *covariates* (observed metadata that influence the distribution of labels and words).

Up to this point, the ability to create variations of LDA such as those listed above has been limited by the expertise needed to develop custom inference algorithms for each model. As a result, it is rare to see such variations being widely used in practice. In this work, I take advantage of recent advances in variational methods (Kingma and Welling, 2014; Rezende et al., 2014; Miao et al., 2016; Srivastava and Sutton, 2017) to facilitate approximate Bayesian inference *without requiring model-specific derivations*, and propose a general neural framework for topic models with metadata, SCHOLAR.[1]

SCHOLAR combines the abilities of SAGE and SLDA, and allows for easy exploration of the following options for customization:

- Covariates: as in SAGE and STM, SCHOLAR can incorporate explicit deviations for observed covariates, as well as effects for *interactions* with topics.
- Supervision: as in SLDA, SCHOLAR can use metadata as labels to help infer topics that are relevant in predicting those labels.
- Rich encoder network: Using the encoding network of a variational autoencoder

[1] **S**parse **C**ontextual **H**idden and **O**bserved **L**anguage **A**utoencode**R**.

(VAE), it is possible to incorporate additional prior knowledge in the form of word embeddings, and/or to provide interpretable embeddings of covariates.

- Sparsity: as in SAGE, a sparsity-inducing prior can be used to encourage more interpretable topics, represented as sparse deviations from a background log-frequency.

I begin with the necessary background and motivation (§4.2), and then describe the basic framework and its extensions (§4.3), followed by a series of experiments (§4.5). In an unsupervised setting, it is possible to customize the model to trade off between perplexity, coherence, and sparsity, with improved coherence through the introduction of word vectors. Alternatively, by incorporating metadata the model can either learn topics that are more predictive of labels than SLDA, or learn explicit deviations for particular parts of the metadata. Finally, by combining all parts of our model SCHOLAR can meaningfully incorporate metadata in multiple ways, which I demonstrate through an exploration of the corpus of immigration articles in the Media Frames Corpus (MFC; see §2.4).

Like more familiar topic models, SCHOLAR can be used for both discovery and measurement, and offers attractive properties in terms of its interpretability. In presenting this particular model, I emphasize not only its ability to adapt to the characteristics of the data, but the extent to which the VAE approach to inference provides a powerful framework for latent variable modeling that suggests the possibility of many further extensions. As such, VAEs have the ability to expand the range of modeling options available to scholars in other disciplines, without requiring the expertise necessary to derive specialized inference algorithms.

4.2 Background and motivation

LDA can be understood as a non-negative Bayesian matrix factorization model: the observed document-word frequency matrix, $\mathbf{X} \in \mathbb{Z}^{D \times V}$ (D is the number of documents, V is the vocabulary size) is factored into two low-rank matrices, $\mathbf{\Theta}^{D \times K}$ and $\mathbf{B}^{K \times V}$, where each row of $\mathbf{\Theta}$, $\boldsymbol{\theta}_i \in \Delta^K$ is a latent variable representing a distribution over topics in document i, and each row of $\mathbf{B}$, $\boldsymbol{\beta}_k \in \Delta^V$, represents a single topic, i.e., a distribution over words in the vocabulary.[2] While it is possible to factor the count data into unconstrained matrices, the particular priors assumed by LDA are important for interpretability (Wallach et al., 2009). For example, the neural variational document model (NVDM; Miao et al., 2016) allows $\boldsymbol{\theta}_i \in \mathbb{R}^K$ and achieves normalization by taking the softmax of $\boldsymbol{\theta}_i^\top \mathbf{B}$. However, the experiments in Srivastava and Sutton (2017) found the performance of the NVDM to be slightly worse than LDA in terms of perplexity, and dramatically worse in terms of topic coherence.

The topics discovered by LDA tend to be parsimonious and coherent groupings of words which are readily identifiable to humans as being related to each other (Chang et al., 2009), and the resulting mode of the matrix $\mathbf{\Theta}$ provides a representation of each document which can be treated as a measurement for downstream tasks, such as classification or answering social scientific questions (Wallach, 2016). LDA does not require — and cannot make use of — additional prior knowledge. As such, the topics that are discovered may bear little connection to metadata of a corpus that is of interest to a researcher, such as sentiment, ideology, or time.

In this chapter, I take inspiration from two models which have sought to alleviate this problem. The first, supervised LDA (SLDA; McAuliffe and Blei, 2008), assumes

[2] $\mathbb{Z}$ denotes nonnegative integers, and Δ^K denotes the set of K-length nonnegative vectors that sum to one. For a proper probabilistic interpretation, the matrix to be factored is actually the matrix of latent mean parameters of the assumed data generating process, $\mathbf{X}_{ij} \sim \text{Poisson}(\mathbf{\Lambda}_{ij})$. See Cemgil (2009) or Paisley et al. (2014) for details.

that documents have labels y which are generated conditional on the corresponding latent representation, i.e., $y_i \sim p(y \mid \boldsymbol{\theta}_i)$.[3] By incorporating labels into the model, it is forced to learn topics which allow documents to be represented in a way that is useful for the classification task. Such models can be used inductively as text classifiers (Balasubramanyan et al., 2012).

SAGE (Eisenstein et al., 2011), by contrast, is an exponential-family model, where the key innovation was to replace topics with sparse deviations from the background log-frequency of words ($\boldsymbol{d}$), i.e., $p(\text{word} \mid \text{softmax}(\boldsymbol{d} + \boldsymbol{\theta}_i^\top \mathbf{B}))$. SAGE can also incorporate deviations for observed covariates, as well as interactions between topics and covariates, by including additional terms inside the softmax. In principle, this allows for inferring, for example, the effect on an author's ideology on their choice of words, as well as ideological variations on each underlying topic. Unlike the NVDM, SAGE still constrains $\boldsymbol{\theta}_i$ to lie on the simplex, as in LDA.

SLDA and SAGE provide two different ways that users might wish to incorporate prior knowledge as a way of guiding the discovery of topics in a corpus: SLDA incorporates labels through a distribution conditional on topics; SAGE includes explicit sparse deviations for each unique value of a covariate, in addition to topics.[4]

Because of the Dirichlet-multinomial conjugacy in the original model, efficient inference algorithms exist for LDA. Each variation of LDA, however, has required the derivation of a custom inference algorithm, which is a time-consuming and error-prone process. In SLDA, for example, each type of distribution one might assume for $p(y \mid \boldsymbol{\theta})$ would require a modification of the inference algorithm. SAGE breaks conjugacy, and

[3] Technically, the model conditions on the mean of the per-word latent variables, but I elide this detail in the interest of concision.

[4] A third way of incorporating metadata is the approach used by various "upstream" models, such as Dirichlet-multinomial regression (Mimno and McCallum, 2008), which uses observed metadata to inform the document prior. I hypothesize that this approach could be productively combined with the SCHOLAR framework, but leave this as future work.

as such, the authors adopted L-BFGS for optimizing the variational bound. Moreover, in order to maintain computational efficiency, it assumed that covariates were limited to a single categorical label.

More recently, the variational autoencoder (VAE) was introduced as a way to perform approximate posterior inference on models with otherwise intractable posteriors (Kingma and Welling, 2014; Rezende et al., 2014). This approach has previously been applied to models of text by Miao et al. (2016) and Srivastava and Sutton (2017). I build on their work and show how this framework can be adapted to seamlessly incorporate the ideas of both SAGE and SLDA, while allowing for greater flexibility in the use of metadata. Moreover, by exploiting automatic differentiation, I allow for modification of the model without requiring any change to the inference procedure. The result is not only a highly adaptable family of models with scalable inference and efficient prediction; it also points the way to incorporation of many ideas found in the literature, such as a temporal evolution of topics (Blei and Lafferty, 2006), and hierarchical models (Blei et al., 2010; Nguyen et al., 2013, 2015c).

4.3 Scholar: A neural topic model with covariates, supervision, and sparsity

I begin by presenting the generative story for SCHOLAR, and explain how it generalizes both SLDA and SAGE (§4.3.1). I then provide a general explanation of inference using VAEs and how it applies to my model (§4.4), as well as how to infer document representations and predict labels at test time (§4.4.1). Finally, I discuss how one can incorporate additional prior knowledge (§4.4.2).

4.3.1 Generative story

Consider a corpus of D documents, where document i is a list of N_i words, $\boldsymbol{w}_i$, with V words in the vocabulary. For each document, one may have observed covariates $\boldsymbol{c}_i$ (e.g., year of publication), and/or one or more labels, $\boldsymbol{y}_i$ (e.g., sentiment).

My model builds on the generative story of LDA, but optionally incorporates labels and covariates, and replaces the matrix product of $\boldsymbol{\Theta}$ and $\mathbf{B}$ with a more flexible generative network, f_g, followed by a softmax transform. Instead of using a Dirichlet prior as in LDA, I employ a logistic normal prior on $\boldsymbol{\theta}$ as in Srivastava and Sutton (2017) to facilitate inference (§4.4): I draw a latent variable, $\boldsymbol{r}$, from a multivariate normal, and transform it to lie on the simplex using a softmax transform.[5]

The generative story is shown in Figure 4.1a and described in equations below:

For each document i of length N_i:

Draw a latent representation on the simplex from a logistic normal prior:

- $\boldsymbol{r}_i \sim \mathcal{N}(\boldsymbol{r} \mid \boldsymbol{\mu}_0(\alpha), \text{diag}(\boldsymbol{\sigma}_0^2(\alpha)))$
- $\boldsymbol{\theta}_i = \text{softmax}(\boldsymbol{r}_i)$

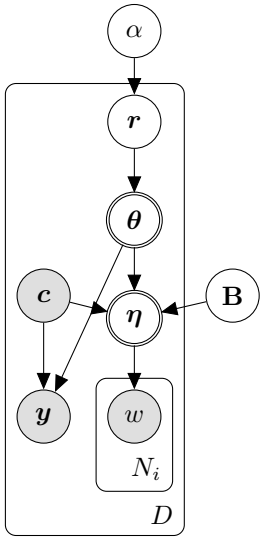
Generate words, incorporating covariates:

- $\boldsymbol{\eta}_i = f_g(\boldsymbol{\theta}_i, \boldsymbol{c}_i)$
- For each word j in document i:
  - $w_{ij} \sim p(w \mid \text{softmax}(\boldsymbol{\eta}_i))$

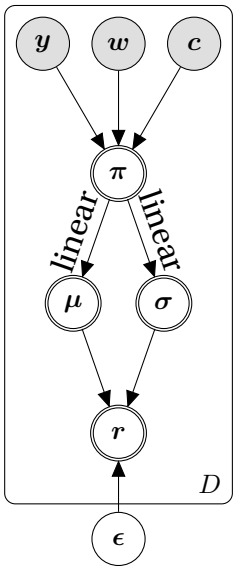
Similarly generate labels:

- $\boldsymbol{y}_i \sim p(\boldsymbol{y} \mid f_y(\boldsymbol{\theta}_i, \boldsymbol{c}_i))$

[5] Unlike the correlated topic model (CTM; Lafferty and Blei, 2006), which also uses a logistic-normal prior, I fix the parameters of the prior and use a diagonal covariance matrix, rather than trying to infer correlations among topics. However, it would be a straightforward extension of this framework to place a richer prior on the latent document representations, and learn correlations by updating the parameters of this prior after each epoch, analogously to the variational EM approach used for the CTM.



(a) Generative model



(b) Inference model

Figure 4.1: (a) presents the generative story of SCHOLAR. (b) illustrates the inference network using the reparametrization trick to perform variational inference on the model presented in this chapter. Shaded nodes are observed; double circles indicate deterministic transformations of parent nodes.

where $p(w \mid \text{softmax}(\boldsymbol{\eta}_i))$ is a multinomial distribution and $p(\boldsymbol{y} \mid f_y(\boldsymbol{\theta}_i, \boldsymbol{c}_i))$ is a distribution appropriate to the data (e.g., multinomial for categorical labels). f_g is a model-specific combination of latent variables and covariates, f_y is a multi-layer neural network, and $\boldsymbol{\mu}_0(\alpha)$ and $\boldsymbol{\sigma}_0^2(\alpha)$ are the mean and diagonal covariance terms of a multivariate normal prior. To approximate a symmetric Dirichlet prior with hyperparameter α, these are given by the Laplace approximation (Hennig et al., 2012) to be $\mu_{0,k}(\alpha) = 0$ and $\sigma_{0,k}^2 = (K-1)/(\alpha K)$.

If one were to ignore covariates, place a Dirichlet prior on $\boldsymbol{\beta}$, and let $\boldsymbol{\eta} = \boldsymbol{\theta}_i^\top \mathbf{B}$, this model is equivalent to SLDA with a logistic normal prior. Similarly, one can recover a model that is like SAGE, but lacks sparsity, if one ignores labels, and let

$$\boldsymbol{\eta}_i = \boldsymbol{d} + \boldsymbol{\theta}_i^\top \mathbf{B} + \boldsymbol{c}_i^\top \mathbf{B}^{cov} + (\boldsymbol{\theta}_i \otimes \boldsymbol{c}_i)^\top \mathbf{B}^{int}, \tag{4.1}$$

where $\boldsymbol{d}$ is the V-dimensional background term (representing the log of the overall word frequency), $\boldsymbol{\theta}_i \otimes \boldsymbol{c}_i$ is a vector of interactions between topics and covariates, and $\mathbf{B}^{cov}$ and $\mathbf{B}^{int}$ are additional weight (deviation) matrices. The background is included to account for common words with approximately the same frequency across documents, meaning that the $\mathbf{B}^*$ weights now represent both positive and negative deviations from this background. This is the form of f_g which I will use in my experiments.

To recover the full SAGE model, one can place a sparsity-inducing prior on each $\mathbf{B}^*$. As in Eisenstein et al. (2011), I make use of the compound normal-exponential prior for each element of the weight matrices, $\mathbf{B}^*_{m,n}$, with hyperparameter γ,[6]

$$\tau_{m,n} \sim \text{Exponential}(\gamma), \tag{4.2}$$

$$\mathbf{B}^*_{m,n} \sim \mathcal{N}(0, \tau_{m,n}). \tag{4.3}$$

One can choose to ignore various parts of this model, if, for example, one doesn't have any labels or observed covariates, or doesn't wish to use interactions or sparsity.[7] Other generator networks could also be considered, with additional layers to represent more complex interactions, although this might involve some loss of interpretability.

In the absence of metadata, and without sparsity, this model is equivalent to the ProdLDA model of Srivastava and Sutton (2017) with an explicit background term, and ProdLDA is, in turn, a special case of SAGE, without background log-frequencies, sparsity, covariates, or labels. In the next section I generalize the inference method used for ProdLDA; in my experiments I validate its performance and explore the effects of regu-

[6] To avoid having to tune γ, I employ an improper Jeffery's prior, $p(\tau_{m,n}) \propto 1/\tau_{m,n}$, as in SAGE. Although this causes difficulties in posterior inference for the variance terms, τ, in practice, I resort to a variational EM approach, with MAP-estimation for the weights, $\mathbf{B}$, and thus alternate between computing expectations of the τ parameters, and updating all other parameters using some variant of stochastic gradient descent. For this, I only require the expectation of each τ_{mn} for each E-step, which is given by $1/\mathbf{B}^2_{m,n}$. I refer the reader to Eisenstein et al. (2011) for additional details.

[7] One could also ignore latent topics, in which case one would get a naive Bayes-like model of text with deviations for each covariate $p(\boldsymbol{w}_{ij} \mid \boldsymbol{c}_i) \propto \exp(\boldsymbol{d} + \boldsymbol{c}_i^\top \mathbf{B}^{cov})$.

larization and word-vector initialization (§4.4.2). The NVDM (Miao et al., 2016) uses the same approach to inference, but does not not restrict document representations to the simplex.

4.4 Learning and inference

As in past work, each document i is assumed to have a latent representation $\boldsymbol{r}_i$, which can be interpreted as its relative membership in each topic (after exponentiating and normalizing). In order to infer an approximate posterior distribution over $\boldsymbol{r}_i$, I adopt the sampling-based VAE framework developed in previous work (Kingma and Welling, 2014; Rezende et al., 2014).

As in conventional variational inference, I assume a variational approximation to the posterior, $q_{\Phi}(\boldsymbol{r}_i \mid \boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i)$, and seek to minimize the KL divergence between it and the true posterior, $p(\boldsymbol{r}_i \mid \boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i)$, where Φ is the set of variational parameters to be defined below. After some manipulations, I obtain the evidence lower bound (ELBO) for a single document,

$$\mathcal{L}(\boldsymbol{w}_i) = \mathbb{E}_{q_{\Phi}(\boldsymbol{r}_i|\boldsymbol{w}_i,\boldsymbol{c}_i,\boldsymbol{y}_i)}\left[\sum_{j=1}^{N_i} \log p(w_{ij} \mid \boldsymbol{r}_i, \boldsymbol{c}_i)\right] + \mathbb{E}_{q_{\Phi}(\boldsymbol{r}_i|\boldsymbol{w}_i,\boldsymbol{c}_i,\boldsymbol{y}_i)}\left[\log p(\boldsymbol{y}_i \mid \boldsymbol{r}_i, \boldsymbol{c}_i)\right] - \mathrm{D}_{\mathrm{KL}}\left[q_{\Phi}(\boldsymbol{r}_i \mid \boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i) \,||\, p(\boldsymbol{r}_i \mid \alpha)\right]. \quad (4.4)$$

As in the original VAE, I will encode the parameters of the variational distributions using a shared multi-layer neural network. Because I have assumed a diagonal normal prior on $\boldsymbol{r}$, this will take the form of a network which outputs a mean vector, $\boldsymbol{\mu}_i = f_{\boldsymbol{\mu}}(\boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i)$ and diagonal of a covariance matrix, $\boldsymbol{\sigma}_i^2 = f_{\boldsymbol{\sigma}}(\boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i)$, such that $q_{\Phi}(\boldsymbol{r}_i \mid \boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$. Incorporating labels and covariates to the inference network

used by Miao et al. (2016) and Srivastava and Sutton (2017), I use:

$$\boldsymbol{\pi}_i = f_e([\mathbf{W}_x \boldsymbol{x}_i; \mathbf{W}_c \boldsymbol{c}_i; \mathbf{W}_y \boldsymbol{y}_i]), \tag{4.5}$$

$$\boldsymbol{\mu}_i = \mathbf{W}_\mu \boldsymbol{\pi}_i + \boldsymbol{b}_\mu, \tag{4.6}$$

$$\log \boldsymbol{\sigma}_i^2 = \mathbf{W}_\sigma \boldsymbol{\pi}_i + \boldsymbol{b}_\sigma, \tag{4.7}$$

where $\boldsymbol{x}_i$ is a V-dimensional vector representing the counts of words in $\boldsymbol{w}_i$, and f_e is a multilayer perceptron. The full set of encoder parameters, $\boldsymbol{\Phi}$, thus includes the parameters of f_e and all weight matrices and bias vectors in Equations 4.5–4.7 (see Figure 4.1b).

This approach means that the expectations in Equation 4.4 are intractable, but one can approximate them using sampling. In order to maintain differentiability with respect to $\boldsymbol{\Phi}$, even after sampling, I make use of the reparameterization trick (Kingma and Welling, 2014),[8] which allows us to reparameterize samples from $q_{\boldsymbol{\Phi}}(\boldsymbol{r} \mid \boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i)$ in terms of samples from an independent source of noise, i.e.,

$$\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}(0, \mathbf{I}), \tag{4.8}$$

$$\boldsymbol{r}_i^{(s)} = g_{\boldsymbol{\Phi}}(\boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i, \boldsymbol{\epsilon}^{(s)}) = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \cdot \boldsymbol{\epsilon}^{(s)}. \tag{4.9}$$

I thus replace the bound in Equation 4.4 with a Monte Carlo approximation using a single sample[9] of $\boldsymbol{\epsilon}$ (and thereby of $\boldsymbol{r}$):

$$\mathcal{L}(\boldsymbol{w}_i) \approx \sum_{j=1}^{N_i} \log p(w_{ij} \mid \boldsymbol{r}_i^{(s)}, \boldsymbol{c}_i) + \log p(\boldsymbol{y}_i \mid \boldsymbol{r}_i^{(s)}, \boldsymbol{c}_i) - \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\Phi}}(\boldsymbol{r}_i \mid \boldsymbol{w}_i, \boldsymbol{c}_i, \boldsymbol{y}_i) \,||\, p(\boldsymbol{r}_i \mid \alpha)\right]. \tag{4.10}$$

[8] The Dirichlet distribution cannot be directly reparameterized in this way, which is why I use the logistic normal prior on $\boldsymbol{\theta}$ to approximate the Dirichlet prior used in LDA.

[9] Alternatively, one can average over multiple samples, either naively or using importance weighting (Burda et al., 2016).

It is now possible to optimize this sampling-based approximation of the variational bound with respect to Φ, B^*, and all parameters of f_g and f_y using stochastic gradient descent. Moreover, because of this stochastic approach to inference, one is *not* restricted to covariates with a small number of unique values, which was a limitation of SAGE. Finally, the KL divergence term in Equation 4.10 can be computed in closed form (see Kingma and Welling, 2014 or Card et al., 2018 for details).

4.4.1 Prediction on held-out data

In addition to inferring latent topics, SCHOLAR can both infer latent representations for new documents and predict their labels, the latter of which was the motivation for SLDA. In traditional variational inference, inference at test time requires fixing global parameters (topics), and optimizing the per-document variational parameters for the test set. With the VAE framework, by contrast, the encoder network (Equations 4.5–4.7) can be used to directly estimate the posterior distribution for each test document, using only a forward pass (no iterative optimization or sampling).

If not using labels, one can use this approach directly, passing the word counts of new documents through the encoder to get a posterior $q_\Phi(\boldsymbol{r}_i \mid \boldsymbol{w}_i, \boldsymbol{c}_i)$. When also including labels to be predicted, one can first train a fully-observed model, as above, then fix the decoder, and retrain the encoder without labels. In practice, however, if one trains the encoder network using one-hot encodings of document labels, it is sufficient to provide a vector of all zeros for the labels of test documents; this is what I adopt for my experiments (§4.5.2), and I still obtain good predictive performance.

The label network, f_y, is a flexible component which can be used to predict a wide range of outcomes, from categorical labels (such as star ratings; McAuliffe and Blei, 2008) to real-valued outputs (such as number of citations or box-office returns;

Yogatama et al., 2011). For categorical labels, predictions are given by

$$\hat{y}_i = \underset{y \in \mathcal{Y}}{\text{argmax}}\, p(y \mid \boldsymbol{\theta}_i, \boldsymbol{c}_i). \tag{4.11}$$

Alternatively, when dealing with a small set of categorical labels, it is also possible to treat them as observed categorical covariates during training. At test time, one can then consider all possible one-hot vectors, $\boldsymbol{e}$, in place of $\boldsymbol{c}_i$, and predict the label that maximizes the probability of the words, i.e.,

$$\hat{y}_i = \underset{y \in \mathcal{Y}}{\text{argmax}} \sum_{j=1}^{N_i} \log p(w_{ij} \mid \boldsymbol{\theta}_i, \boldsymbol{e}_y). \tag{4.12}$$

This approach works well in practice (as I show in §4.5.2), but does not scale to large numbers of labels, or other types of prediction problems, such as multi-class classification or regression.

The choice to include metadata as covariates, labels, or both, depends on the data. The key point is that one can incorporate metadata in two very different ways, depending on what one wants from the model. Labels guide the model to infer topics that are relevant to those labels, whereas covariates induce explicit deviations, leaving the latent variables to account for the rest of the content.

4.4.2 Additional prior information

A final advantage of the VAE framework is that the encoder network provides a way to incorporate additional prior information in the form of word vectors. Although one can learn all parameters starting from a random initialization, it is also possible to initialize and fix the initial embeddings of words in the model, $\mathbf{W}_x$, in Equation 4.5. This leverages word similarities derived from large amounts of unlabeled data, and may

promote greater coherence in inferred topics. The same could also be done for some covariates; for example, one could embed the source of a news article based on its place on the ideological spectrum. Conversely, if one chooses to learn these parameters, the learned values ($\mathbf{W}_y$ and $\mathbf{W}_c$) may provide meaningful embeddings of these metadata (see section §4.5.3).

Other variants on topic models have also proposed incorporating word vectors, both as a parallel part of the generative process (Nguyen et al., 2015b), and as an alternative parameterization of topic distributions (Das et al., 2015), but inference is not scalable in either of these models. Because of the generality of the VAE framework, one could also modify the generative story so that word embeddings are emitted (rather than tokens); I leave this for future work.

4.4.3 Additional details

As observed in past work, inference using a VAE can suffer from component collapse, which translates into excessive redundancy in topics (i.e., many topics containing the same set of words). To mitigate this problem, I borrow the approach used by Srivastava and Sutton (2017), and make use of the Adam optimizer with a high momentum, combined with batchnorm layers to avoid divergence. Specifically, I add batchnorm layers following the computation of $\boldsymbol{\mu}$, $\log \boldsymbol{\sigma}^2$, and $\boldsymbol{\eta}$.

This effectively solves the problem of mode collapse, but the batchnorm layer on $\boldsymbol{\eta}$ introduces an additional problem, not reported in past work. At test time, the batchnorm layer will shift the input based on the learned population mean of the training data; this effectively encodes information about the distribution of words in this model that is not captured by the topic weights and background distribution. As such, although reconstruction error will be low, the document representation $\boldsymbol{\theta}$, will

not necessarily be a useful representation of the topical content of each document. In order to alleviate this problem, I reconstruct η as a convex combination of two copies of the output of the generator network, one passed through a batchnorm layer, and one not. During training, I then gradually anneal the model from relying entirely on the component passed through the batchnorm layer, to relying entirely on the one that is not. This ensures that the the final weights and document representations will be properly interpretable.

4.5 Experiments and results

To evaluate and demonstrate the potential of this model, I present a series of experiments below. I first test SCHOLAR without observed metadata, and explore the effects of using regularization and/or word vector initialization, compared to LDA, SAGE, and NVDM (§4.5.1). I then evaluate my model in terms of predictive performance, in comparison to SLDA and an l_2-regularized logistic regression baseline (§4.5.2). Finally, I demonstrate the ability to incorporate covariates and/or labels in an exploratory data analysis (§4.5.3).

The scores I report are generalization to held-out data, measured in terms of perplexity; coherence, measured in terms of non-negative point-wise mutual information (NPMI; Chang et al., 2009; Newman et al., 2010), and classification accuracy on test data. For coherence I evaluate NPMI using the top 10 words of each topic, both internally (using test data), and externally, using a decade of articles from the English Gigaword dataset (Graff and Cieri, 2003). Since SCHOLAR employs variational methods, the reported perplexity is an upper bound based on the ELBO.

As datasets I use the familiar 20 newsgroups, the IMDB corpus of 50,000 movie reviews (Maas et al., 2011), and the UIUC Yahoo answers dataset with 150,000 docu-

ments in 15 categories (Chang et al., 2008). For further exploration, I also make use of the immigration articles from the MFC, each annotated with pro- or anti-immigration tone (see Chapter 2). All datasets were preprocessed by tokenizing, converting to lower case, removing punctuation, and dropping all tokens that included numbers, all tokens less than 3 characters, and all words on the stopword list from the snowball sampler.[10] The vocabulary was then formed by keeping the words that occurred in the most documents.

For all experiments I use a model with 300-dimensional embeddings of words, and take f_e to be the element-wise softplus nonlinearity (followed by the linear transformations for $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$). Similarly, f_y is a linear transformation of $\boldsymbol{\theta}$, followed by a softplus layer, followed by a linear transformation to the size of the output (the number of classes). During training, I set S (the number of samples of $\boldsymbol{\epsilon}$) to be 1; for estimating the ELBO at on test documents, I set $S = 20$.

I use the original author-provided implementations of SAGE[11] and SLDA,[12] while for LDA I use Mallet.[13] My implementation of SCHOLAR for these experiments was in TensorFlow, but I have also provided a PyTorch implementation of the core of the model.[14]

It is challenging to fairly evaluate the relative computational efficiency of this approach compared to past work (due to the stochastic inference, choices about hyperparameters such as tolerance, and because of differences in implementation). Nevertheless, in practice, the performance of SCHOLAR is highly appealing. For all experiments in this chapter, my implementation was much faster than SLDA or SAGE (implemented in C and Matlab, respectively), and competitive with Mallet.

[10] snowball.tartarus.org/algorithms/english/stop.txt
[11] github.com/jacobeisenstein/SAGE
[12] github.com/blei-lab/class-slda
[13] mallet.cs.umass.edu
[14] github.com/dallascard/scholar

4.5.1 Unsupervised evaluation

Although the emphasis of this work is on incorporating observed labels and/or covariates, I briefly report on experiments in the unsupervised setting. Recall that, without metadata, SCHOLAR equates to ProdLDA, but with an explicit background term. I therefore use the same experimental setup as Srivastava and Sutton (2017) (learning rate, momentum, batch size, and number of epochs) and find the same general patterns as they reported (see Tables 4.1, 4.2, and 4.3): in general, SCHOLAR returns more coherent topics than LDA, but at the cost of worse perplexity. SAGE, by contrast, attains very high levels of sparsity, but at the cost of worse perplexity and coherence than LDA. As expected, the NVDM produces relatively low perplexity, but very poor coherence, due to its lack of constraints on θ.

Further experimentation revealed that the VAE framework involves a tradeoff among the scores; running for more epochs tends to result in better perplexity on held-out data, but at the cost of worse coherence. This phenomenon has recently been investigated in more detail, revealing that while both perplexity and coherence tend to improve early on, at some point coherence will begin to decrease (sometimes catastrophically), while perplexity will continue to improve (Ding et al., 2018). Recent work building on the model presented in this chapter noted the same finding, and proposed using coherence (as measured by NPMI) as a criterion for early stopping (Gururangan et al., 2019).

Adding regularization to encourage sparse topics has a similar effect as in SAGE, leading to worse perplexity and coherence, but it does create sparse topics. Interestingly, initializing the encoder with pretrained word2vec embeddings, and *not* updating them tends to produce a model with the best internal coherence on most datasets.

Finally, the background term in SCHOLAR model does not have much effect on perplexity, but plays an important role in producing coherent topics; as in SAGE, the

Model	Ppl. ↓	NPMI (int.) ↑	NPMI (ext.) ↑	Sparsity ↑
LDA	**1508**	0.13	0.14	0
SAGE	1767	0.12	0.12	**0.79**
NVDM	1748	0.06	0.04	0
SCHOLAR − B.G.	1889	0.09	0.13	0
SCHOLAR	1905	0.14	0.13	0
SCHOLAR + W.V.	1991	**0.18**	**0.17**	0
SCHOLAR + REG.	2185	0.10	0.12	0.58

Table 4.1: Performance of the various models in an unsupervised setting (i.e., without labels or covariates) on the IMDB dataset using a 5,000-word vocabulary and 50 topics. The best result in each column is shown in bold.

Model	Ppl. ↓	NPMI (int.) ↑	NPMI (ext.) ↑	Sparsity ↑
LDA	**810**	0.20	0.11	0
SAGE	867	0.27	0.15	**0.71**
NVDM	1067	0.18	0.11	0
SCHOLAR - B.G.	928	0.17	0.09	0
SCHOLAR	921	**0.35**	0.16	0
SCHOLAR + W.V.	955	0.29	**0.17**	0
SCHOLAR + REG.	1053	0.25	0.13	0.43

Table 4.2: Performance of various models on the 20 newsgroups dataset with 20 topics and a 2,000-word vocabulary.

background can account for common words, so they are mostly absent among the most heavily weighted words in the topics. For instance, words like *film* and *movie* in the IMDB corpus are relatively unimportant in the topics learned by SCHOLAR model, but would be much more heavily weighted without the background term, as they are in topics learned by LDA. Moreover, the background term also helps to avoids some of the repetition among top terms in topics that can otherwise occur.

An example of 20 topics learned by SCHOLAR on the 20 newsgroups dataset is shown in Table 4.4

Model	Ppl. ↓	NPMI (int.) ↑	NPMI (ext.) ↑	Sparsity ↑
LDA	**1035**	0.29	0.15	0
NVDM	4588	0.20	0.09	0
SCHOLAR - B.G.	1589	0.27	**0.16**	0
SCHOLAR	1596	0.33	0.13	0
SCHOLAR + W.V.	1780	**0.37**	0.15	0
SCHOLAR + REG.	1840	0.34	0.13	0.44

Table 4.3: Performance of various models on the Yahoo answers dataset with 250 topics and a 5,000-word vocabulary. SAGE did not finish in 72 hours so I omit it from this table.

NPMI	Topic
0.77	turks armenian armenia turkish roads escape soviet muslim mountain soul
0.52	escrow clipper encryption wiretap crypto keys secure chip nsa key
0.49	jesus christ sin bible heaven christians church faith god doctrine
0.43	fbi waco batf clinton children koresh compound atf went fire
0.41	players teams player team season baseball game fans roger league
0.39	guns gun weapons criminals criminal shooting police armed crime defend
0.37	playoff rangers detroit cup wings playoffs montreal toronto minnesota
0.36	ftp images directory library available format archive graphics package
0.33	user server faq archive users ftp unix applications mailing directory
0.32	bike car cars riding ride engine rear bmw driving miles
0.32	study percent sexual medicine gay studies april percentage treatment
0.32	israeli israel arab peace rights policy islamic civil adam citizens
0.30	morality atheist moral belief existence christianity truth exist god objective
0.28	space henry spencer international earth nasa orbit shuttle development
0.27	bus motherboard mhz ram controller port drive card apple mac
0.25	windows screen files button size program error mouse colors microsoft
0.24	sale shipping offer brand condition sell printer monitor items asking
0.21	driver drivers card video max advance vga thanks windows appreciated
0.19	cleveland advance thanks reserve ohio looking nntp western host usa
0.04	banks gordon univ keith soon pittsburgh michael computer article ryan

Table 4.4: Topics from the unsupervised SCHOLAR on the 20 newsgroups dataset, and the corresponding internal coherence values.

	20news	IMDB	Yahoo
Vocabulary size	2000	5000	5000
Number of topics	50	50	250
SLDA	0.60	0.64	0.65
SCHOLAR (labels)	0.67	0.86	0.73
SCHOLAR (covariates)	**0.71**	**0.87**	0.72
Logistic regression	0.70	**0.87**	**0.76**

Table 4.5: Accuracy of various models on three datasets with categorical labels.

4.5.2 Text classification

I next consider the utility of SCHOLAR in the context of categorical labels, and consider them alternately as observed covariates and as labels generated conditional on the latent representation. I use the same setup as above, but tune number of training epochs for this model using a random 20% of training data as a development set, and similarly tune regularization for logistic regression.

Table 4.5 summarizes the accuracy of various models on three datasets, revealing that SCHOLAR offers competitive performance, both as a joint model of words and labels (Eq. 4.11), and a model which conditions on covariates (Eq. 4.12). Although SCHOLAR is comparable to the logistic regression baseline, my purpose here is not to attain state-of-the-art performance on text classification. Rather, the high accuracies I obtain demonstrate that the model are learning low-dimensional representations of documents that are relevant to the label of interest, outperforming SLDA, and have the same attractive properties as topic models. Further, any neural network that is successful for text classification could be incorporated into f_y and trained end-to-end along with topic discovery.

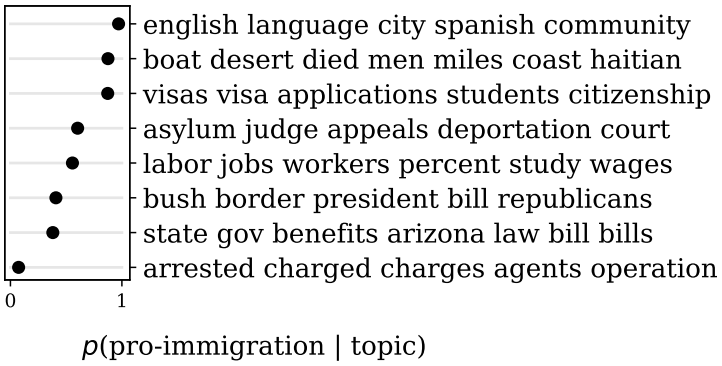


Figure 4.2: Topics inferred by a joint model of words and tone, and the corresponding probability of pro-immigration tone for each topic. A topic is represented by the top words sorted by word probability throughout the chapter.

4.5.3 Exploratory study

In this section, I demonstrate how SCHOLAR might be used to explore an annotated corpus of articles about immigration, and adapt to different assumptions about the data. I only use a small number of topics in this part ($K = 8$) for compact presentation.

Tone as a label. I first consider using the annotations as a *label*, and train a joint model to infer topics relevant to the tone of the article (pro- or anti-immigration). Figure 4.2 shows a set of topics learned in this way, along with the predicted probability of an article being pro-immigration conditioned on the given topic. All topics are coherent, and the predicted probabilities have strong face validity, e.g., "arrested charged charges agents operation" is least associated with pro-immigration.

Tone as a covariate. Next I consider using tone as a *covariate*, and build a model using both tone and tone-topic interactions. Table 4.6 shows a set of topics learned from the immigration data, along with the most highly-weighted words in the corresponding tone-topic interaction terms. As can be seen, these interaction terms tend to capture different frames (e.g., "criminal" vs. "detainees", and "illegals" vs. "newcomers", etc).

Base topics (each row is a topic)	Anti-immigration interactions	Pro-immigration interactions
ice customs agency enforcement	**criminal** customs arrested	**detainees** detention center
population born percent	jobs million **illegals**	english **newcomers** hispanic
judge case court guilty appeals	guilty charges man charged	asylum court judge case appeals
patrol border miles coast desert	patrol border agents boat	died authorities desert border
licenses drivers card visa cards	foreign sept visas system	green citizenship card citizen
island story chinese ellis	smuggling federal charges	island school ellis english story
guest worker workers bush labor	bill border house senate	workers tech skilled farm labor
benefits bill welfare republican	republican california gov	law welfare students tuition

Table 4.6: Top words for topics (left) and the corresponding anti-immigration (middle) and pro-immigration (right) variations when treating tone as a covariate, with interactions.

Combined model with temporal metadata. Finally, I incorporate both the tone annotations and the year of publication of each article, treating the former as a label and the latter as a covariate. In this model, I also include an embedding matrix, $\mathbf{W}_c$, to project the one-hot *year* vectors down to a two-dimensional continuous space, with a learned deviation for each dimension. I omit the topics in the interest of space, but Figure 4.3 shows the learned embedding for each year, along with the top terms of the corresponding deviations. As can be seen, the model learns that adjacent years tend to produce similar deviations, even though I have not explicitly encoded this information. The left-right dimension roughly tracks a temporal trend with positive deviations shifting from the years of "Clinton" and "INS" on the left, to "Obama" and "ICE" on the right.[15] Meanwhile, the events of 9/11 dominate the vertical direction, with the words "sept", "hijackers", and "attacks" increasing in probability as one moves up in the space. One could also to look at each year individually, by dropping the embedding of years, and instead learning a sparse set of topic-year interactions, similar to tone in Table 4.6.

[15]The Immigration and Naturalization Service (INS) was transformed into Immigration and Customs Enforcement (ICE) and other agencies in 2003.

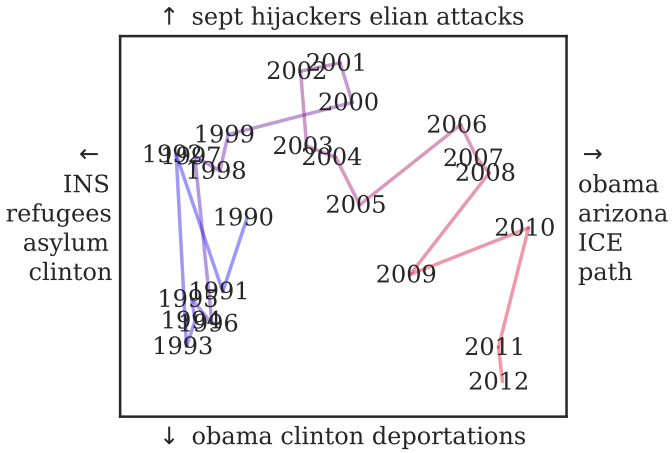


Figure 4.3: Learned embeddings of year-of-publication (treated as a covariate) from combined model of news articles about immigration.

4.6 Additional related work

The literature on topic models is vast; in addition to papers cited throughout, other efforts to incorporate metadata into topic models include Dirichlet-multinomial regression (DMR; Mimno and McCallum, 2008), Labeled LDA (Ramage et al., 2009), and MedLDA (Zhu et al., 2009). A recent paper also extended DMR by using deep neural networks to embed metadata into a richer document prior (Benton and Dredze, 2018). An extension of the work presented in this chapter has also demonstrated how neural variational document models can be effectively used for pretraining document representations for semi-supervised text classification in the low-resource setting (Gururangan et al., 2019).

A separate line of work has pursued parameterizing unsupervised models of documents using neural networks (Hinton and Salakhutdinov, 2009; Larochelle and Lauly, 2012), including non-Bayesian approaches (Cao et al., 2015). More recently, Lau et al. (2017) proposed a neural language model that incorporated topics, and He et al. (2017) developed a scalable alternative to the correlated topic model by simultaneously learning topic embeddings.

Others have attempted to extend the reparameterization trick to the Dirichlet and Gamma distributions, either through transformations (Kucukelbir et al., 2016) or a generalization of reparameterization (Ruiz et al., 2016). Black-box and VAE-style inference have been implemented in at least two general purpose tools designed to allow rapid exploration and evaluation of models (Kucukelbir et al., 2015; Tran et al., 2016).

A natural extension of the work in this chapter would be to develop a neural variational approach to inference in the type of hierarchical model developed for inferring latent personas in Chapter 3. Although there have been several proposed approaches to hierarchical VAEs (Sønderby et al., 2016; Klushyn et al., 2019), these have not yet been fully developed for text data or social science applications more broadly.

4.7 Summary

In this chapter, I have presented a neural framework for generalized topic models to enable flexible incorporation of metadata with a variety of options. I take advantage of stochastic variational inference to develop a general algorithm for this framework such that variations do not require any model-specific algorithm derivations. The resulting model, SCHOLAR, demonstrates the tradeoff between perplexity, coherence, and sparsity, and outperforms SLDA in predicting document labels. Furthermore, this model and accompanying code can facilitate rapid exploration of document collections with metadata, as demonstrated by an example using the immigration articles from the Media Frames Corpus.

Chapter 5

Estimating label proportions from annotations

(This chapter was originally published as Card and Smith, 2018)

5.1 Introduction

As discussed in Chapter 2, a methodological tool often used in the social sciences and humanities (and practical settings like journalism) is *content analysis* – the manual categorization of pieces of text into a set of categories which have been developed to answer a substantive research question (Krippendorff, 2012). *Automated* content analysis holds great promise for augmenting the efforts of human annotators (O'Connor et al., 2011; Grimmer and Stewart, 2013). While this task bears similarity to text categorization problems such as sentiment analysis, the quantity of real interest is often the *proportion* of documents in a dataset that should receive each label (Hopkins and King, 2010). This chapter tackles the problem of estimating label proportions in a target corpus based on a small sample of human annotated data.

As an example, consider the hypothetical question (not explored in this work) of whether hate speech is increasingly prevalent in social media posts in recent years. "Hate speech" is a difficult-to-define category only revealed (at least initially) through human judgments (Davidson et al., 2017; Sap et al., 2019). Note that the goal would not be to identify individual instances, but rather to estimate a proportion, as a way of measuring the prevalence of a social phenomenon. Although I assume that trained annotators could recognize this phenomenon with some acceptable level of agreement, relying solely on manual annotation would restrict the number of messages that could be considered, and would limit the analysis to the messages available at the time of annotation.[1]

I thus treat proportion estimation as a *measurement* problem, and seek a way to train an instrument from a limited number of human annotations to measure label proportions in an unannotated target corpus.

This problem can be cast within a supervised learning framework, and past work has demonstrated that it is possible to improve upon a naive classification-based approach, even without access to any labeled data from the target corpus (Forman, 2005, 2008; Bella et al., 2010; Hopkins and King, 2010; Esuli and Sebastiani, 2015). However, as I argue (§5.2), most of this work is based on a set of assumptions that I believe are invalid in a significant portion of text-based research projects in the social sciences and humanities.

The main contributions of this chapter include:

- identifying two different data-generating scenarios for text data (*intrinsic* vs. *extrinsic* labels) and and establishing their importance to the problem of estimating proportions (§5.2);

[1] For additional examples see Grimmer et al. (2012), Hopkins and King (2010), and references therein.

- analyzing which methods are suitable for each setting, and proposing a simple alternative approach for extrinsic labels (§5.3); and
- an empirical comparison of methods that validates this analysis (§5.4).

Complicating matters somewhat is the fact that annotation may take place before the entire collection is available, so that the subset of instances that are manually annotated may represent a biased sample (§5.2). Because this is so frequently the case, all of the results in this chapter assume that one must confront the challenges of *transfer learning* or *domain adaptation*. (The simpler case, where one can sample from the true population of interest, is revisited in §5.5.)

5.2 Problem definition

The setup in this chapter is similar to that faced in transfer learning, and I will use similar terminology (Pan and Yang, 2010; Weiss et al., 2016). Specifically, I assume that one has a source and a target corpus, comprised of N_S and N_T documents respectively, the latter of which are not available for annotation. I will represent each corpus as a set of documents, i.e., $\boldsymbol{X}^{(S)} = \langle \boldsymbol{x}_1^{(S)}, ..., \boldsymbol{x}_{N_s}^{(S)} \rangle$, and similarly for $\boldsymbol{X}^{(T)}$.

I further assume that one has a set of K mutually exclusive categories, $\mathcal{Y} = \{1, \ldots, K\}$, and that one wishes to estimate the proportion of documents in the target corpus that belong to each category. These would typically correspond to a quantity one wishes to measure, such as what fraction of news articles frame a policy issue in a particular way, what fraction of product reviews are considered helpful, or what fraction of social media messages convey positive sentiment. Generally speaking, these categories will be designed based on theoretical assumptions, an understanding of the design of the platform that produced the data, and/or initial exploration of the data itself.

In idealized text classification scenarios, it is conventional to assume training data with already-assigned gold-standard labels. Here, I are interested in scenarios where one must generate the labels via an annotation process.[2] Specifically, assume that there exists some annotation function, $\mathcal{A}$, which produces a distribution over the K mutually exclusive labels, conditional on text. Given a document, $\boldsymbol{x}_i$, the annotation process samples a label from the annotation function, defined as:

$$\mathcal{A}(\boldsymbol{x}_i, k) \triangleq p(y_i = k \mid \boldsymbol{x}_i). \tag{5.1}$$

Typically, the annotation function would represent the behavior of a human annotator (or group of annotators), but it could also represent a less controlled real-world process, such as users rating a review's helpfulness. Note that this setup *does* include the special case in which true gold-standard labels are available for each instance (such as the authors of documents in an authorship attribution problem). In such a case, $\mathcal{A}$ is deterministic (assuming unique inputs).

Given that my objective is to mimic the annotation process, I seek to estimate the proportion of documents in the target corpus expected to be categorized into each of the K categories, if one had an unlimited budget and full access to the target corpus at the time of annotation. That is, I wish to estimate $q^{(T)}$, which I define as:

$$q(y = k \mid \boldsymbol{X}^{(T)}) \triangleq \tfrac{1}{N_T} \textstyle\sum_{i=1}^{N_T} p(y_i = k \mid \boldsymbol{x}_i^{(T)}). \tag{5.2}$$

Given a set of documents sampled from the *source* corpus and L applications of the annotation function, one can obtain, at some cost, a labeled training corpus of L documents, i.e., $D^{(\text{train})} = \langle (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_L, y_L) \rangle$. Because the source and target cor-

[2] This could include gathering multiple independent annotations per instance, but I will typically assume only one.

Assumption	Intrinsic labels	Extrinsic labels
Data generating process	$\boldsymbol{x} \sim p(\boldsymbol{x} \mid y)$	$y \sim p(y \mid \boldsymbol{x})$
Assumed to differ across domains	$p(y)$	$p(\boldsymbol{x})$
Assumed constant across domains	$p(\boldsymbol{x} \mid y)$	$p(y \mid \boldsymbol{x})$
Corresponding distributional shift	Prior probability shift	Covariate shift

Table 5.1: Data generating scenarios and corresponding distributional properties.

pora are not in general drawn from the same distribution, I seek to make explicit the assumptions about how they differ.[3] Past literature on transfer learning has identified several patterns of dataset shift (Storkey, 2009). Here I focus on two particularly important cases, linking them to the relevant data generating processes, and analyze their relevance to estimating proportions.

Two kinds of distributional shift. There are two natural assumptions one could make about what is constant between the two corpora. One could assume that there is no change in the distribution of text given a document's label, i.e., $p^{(S)}(\boldsymbol{x} \mid y) = p^{(T)}(\boldsymbol{x} \mid y)$. Alternately, one could assume that there is no change in the distribution of labels given text, i.e., $p^{(S)}(y \mid \boldsymbol{x}) = p^{(T)}(y \mid \boldsymbol{x})$. The former is assumed in the case of *prior probability shift*, where one assumes that $p(y)$ differs but $p(\boldsymbol{x} \mid y)$ is constant, and the later is assumed in the case of *covariate shift*, where one assumes that $p(\boldsymbol{x})$ differs but $p(y \mid \boldsymbol{x})$ is constant (Storkey, 2009).

These two assumptions correspond to two fundamentally different types of scenarios that one needs to consider, which are summarized in Table 5.1. The first is where one is dealing with what I will call *intrinsic* labels, that is labels which are inherent to each instance, and which in some sense precede and predict the generation of the text of that instance. A classic example of this scenario is the case of authorship attribution (e.g., Mosteller and Wallace, 1964), in which different authors are assumed to have dif-

[3] Clearly, if one makes no assumptions about how the source and target distributions are related, there is no guarantee that supervised learning will work (Ben-David et al., 2012).

ferent propensities to use different styles and vocabularies. The identity of the author of a document is arguably an intrinsic property of that document, and it is easy to see a text as having been generated conditional on its author.

The contrasting scenario is what I will refer to as *extrinsic* labels; this scenario is my primary interest. I assume here that the labels are not inherent in the documents, but rather have been externally generated, conditional on the text as a stimulus to some behavioral process.[4] I argue that this is the relevant assumption for most annotation-based projects in the social sciences, where the categories of interest do not correspond to pre-existing categories that might have existed in the minds of authors before writing, or affected the writing process. Rather, these are theorized categories that have been developed specifically to analyze or measure some aspect of the document's *effect* that is of interest to the researcher.

One won't always know the true distributional properties of one's datasets, but distinguishing between intrinsic and extrinsic labels provides a guide. The critical point is that these two different labeling scenarios have different implications for robustness to distributional shift. In the case of extrinsic labels, especially when working with trained annotators, it is reasonable to assume that the behavior of the annotation function is determined purely by the text, such that $p(y \mid \boldsymbol{x})$ is unchanged between source and target, and any change in label proportions is explained by a change in the underlying distribution of text, $p(\boldsymbol{x})$. With intrinsic labels, by contrast, it *may* be the case that $p(\boldsymbol{x} \mid y)$ is the same for the source and the target, assuming there are no additional factors influencing the generation of text. In that case, a shift in the distribution of features would be fully explained by a difference in the underlying label proportions.

[4] Fong and Grimmer (2016) also consider this process in attempting to identify the causal effects of texts.

The idea that there are different data generating processes is obviously not new.[5] What is novel here, however, is asking how these different assumptions affect the estimation of proportions. Virtually all past work on estimating proportions has only considered prior probability shift, assuming that $p(\boldsymbol{x} \mid y)$ is constant.[6] Existing methods take advantage of this assumption, and can be shown empirically to work well when it is satisfied (e.g., through artificial modification of real datasets to alter label proportions in a corpus). One would expect them to fail, however, in the case of extrinsic annotations, as there is no reason to think that the required assumption should necessarily hold.

By contrast, the problem of covariate shift is in some sense less of a problem because one directly observes $\boldsymbol{X}^{(T)}$. Since the annotation function is assumed to be unchanging, one could perfectly predict the expected label proportions in the target corpus if one could learn the annotation function using labeled data from the source corpus. The problem thus becomes how to learn a well-calibrated approximation of the annotation function from a limited amount of labeled data.

5.3 Methods

Given a labeled training set and a target corpus, the naive approach is to train a classifier through any conventional means, predict labels on the target corpus, and return the relative prevalence of predicted labels. Following Forman (2005), I refer to this approach as **classify and count** (CC). If using a probabilistic classifier, averaging the predicted posterior probabilities rather than predicted labels will be referred to as **probabilistic classify and count** (PCC; Bella et al., 2010).

Both approaches can fail, however. In the case of intrinsic labels, this is because

[5] Peters et al. (2014) describe these, somewhat confusingly, as *causal* and *anti-causal* problems.

[6] For example, Hopkins and King (2010) argue that bloggers first decide on the sentiment they wish to convey and then write a blog post conditional on that sentiment.

these approaches will not account for the shift in prior label probability, $p(y)$, which is assumed to have occurred (Hopkins and King, 2010). In the case of covariate shift, the difference in $p(\boldsymbol{x})$ will result in a model that is not optimal (in terms of classification performance) for the target domain. In both cases, there is also the problem of classifier bias or miscalibration. Particularly in the case of unbalanced labels, a standard classifier is likely to be biased, overestimating the probability of the more common labels, and vice versa (Zhao et al., 2017). Here I present a simple but novel method for extrinsic labels, followed by a number of baseline approaches against which I will compare.

5.3.1 Proposed method: Calibrated probabilistic classify and count (PCC^{cal})

One simple solution, which I propose here, is to attempt to train a well-calibrated classifier. To be clear, calibration refers to the long-run accuracy of predicted probabilities. That is, a probabilistic classifier, $h_\theta(\boldsymbol{x})$, is well calibrated at the level μ if, among all instances for which the classifier predicts class k with a probability of μ, the proportion that are truly assigned to class k is also equal to μ.[7]

It has previously been shown (DeGroot and Fienberg, 1983; Bröcker, 2009) that any proper scoring rule (e.g., cross entropy, Brier score, etc.) can be factored into two components representing *calibration* and *refinement*, the later of which effectively measures how close predicted probabilities are to zero or one. Minimizing a corresponding loss function thus involves a trade-off between these two components.

Optimizing *only* for calibration is not helpful, as a trivial solution is to simply predict a probability distribution equal to the observed label proportions in the training data for all instances (which is perfectly calibrated on the labeled sample). The alternative

[7] For example, a weather forecaster will be well-calibrated if it rains on 60% of days for which the forecaster predicted a 60% chance of rain, etc.

I propose here is to train a classifier using a typical objective (here, regularized log loss) but use *calibration on held-out data* as a criterion for *model selection*, i.e., when one tunes hyperparameters via cross validation. I refer to this method as **calibrated PCC** (PCC^{cal}). Specifically, I propose to select regularization strength via grid search, choosing the value that leads to the lowest average calibration error on held-out data, averaging over splits of the data. Of course, other hyperparameters could be included in model selection as well.

To estimate calibration error (CE) during cross-validation, I use an approximation due to Nguyen and O'Connor (2015), adaptive binning. In the case of binary labels, this is computed as:

$$\text{CE} \triangleq \frac{1}{B} \sum_{j=1}^{B} \left(\frac{1}{|\mathcal{B}_j|} \sum_{i \in \mathcal{B}_j} y_i - p_\theta(\boldsymbol{x}_i) \right)^2, \tag{5.3}$$

using B bins, where bin $\mathcal{B}_j$ contains instances for which $p_\theta(\boldsymbol{x}_i)$ are in the jth quantile, where $p_\theta(\boldsymbol{x}_i)$ is the predicted probability of a positive label for instance i. For added robustness, I take the average of CE for $B \in \{3, 4, 5, 6, 7\}$.

In my experiments, I consider two variants of PCC: the first, PCC^{F_1}, which is a baseline, is tuned conventionally for classification performance, whereas the other (PCC$^{\text{cal}}$) is tuned for calibration, as measured using CE, but is otherwise identically trained. As a base classifier I make use of l_1-regularized logistic regression, operating on n-gram features.[8]

5.3.2 Existing methods appropriate for extrinsic labels

The idea of extrinsic labels has not been previously considered by past work on estimating proportions, but it is closely related to the problems of calibration and covariate

[8] More complex models could be considered, but I use logistic regression because it is a well-understood and widely applicable model that has been shown to be relatively well-calibrated in general (Niculescu-Mizil and Caruana, 2005).

shift. Here I briefly summarize two representative methods, which I consider as baselines.

Platt scaling. One approach to calibration is to train a model using conventional methods and to then learn a secondary calibration model. One of the most common and successful variations on this approach is Platt scaling, which learns a logistic regression classifier on held-out training data, taking the scores from the primary classifier as input. This model is then applied to the scores returned by the primary classifier on the target corpus (Platt, 1999). To estimate proportions, the predicted probabilities are then averaged, as in PCC.

Reweighting for covariate shift. Although they are not typically thought of in the context of estimating proportions, several methods have been proposed to deal directly with the problem of covariate shift, including kernel mean matching and its extensions (Huang et al., 2006; Sugiyama et al., 2011). Here, I consider the two-stage method from Bickel et al. (2009), which uses a logistic regression model to distinguish between source and target domains, and then uses the probabilities from this model to re-weight labeled training instances, to more heavily favor those that are representative of the target domain. The appeal of this method is that all unlabeled data can be used to estimate this shift.

5.3.3 Existing methods appropriate for intrinsic labels

As previously mentioned, virtually all of the past work on estimating proportions makes the assumption that $p(\boldsymbol{x} \mid y)$ is constant between source and target. Under this assumption, it can be shown that $p(y^{(\theta)} = j \mid y = k)$ is also constant for all j and k, where $y^{(\theta)}$ is the predicted label from h_θ, and y is the true (intrinsic) label. If these

values were known, then the label proportions in the target corpus could be found by taking the model's estimate of label proportions in the target corpus, (CC), and then solving a linear system of equations as a post-classification correction. Although a number of variations on this model have been proposed, all are based on the same assumption, thus I take a method known as **adjusted classify and count** (ACC) as an exemplar, which directly estimates the relevant quantities using a confusion matrix (Forman, 2005). In the case of binary classification, this reduces to:

$$\hat{q}_{\text{ACC}}(y = 1 \mid \boldsymbol{X}^{(T)}) = \frac{\frac{1}{N_T}\sum_{i=1}^{N_T} y_i^{(\theta)} - \text{FPR}}{\text{TPR} - \text{FPR}}, \tag{5.4}$$

where $\text{FPR} = \hat{p}(y^{(\theta)} = 1 \mid y = 0)$ and $\text{TPR} = \hat{p}(y^{(\theta)} = 1 \mid y = 1)$ are both estimated using held-out data. Because ACC can result in inadmissible values in extreme cases, I threshold its predictions to be in the range [0, 1].

5.4 Experiments

For the experiments in this chapter, I focus on the case of binary classification where the difference between the source and target corpora results from a difference in time—that is, the training documents are sampled from one time period, and the goal is to estimate label proportions on documents from a future time period. I include examples of both intrinsic and extrinsic labels to demonstrate the importance of this distinction to the effectiveness of different methods.

As described below, I create multiple subtasks from each dataset by using different partitions of the data. In all cases, I report absolute error (AE) on the proportion of positive instances, averaged across the subtasks of each dataset.

Although I do not have access to the true annotation function, I approximate the expected label proportions in the target corpus by averaging the available labels, which should be a very close approximation when the number of available labels is large (which informed my choice of datasets for these experiments). For a single subtask, the absolute error is thus evaluated as

$$\text{AE} = \left| \hat{q}(y = 1 \mid \boldsymbol{X}^{(T)}) - \tfrac{1}{N_T} \textstyle\sum_{i=1}^{N_T} y_i^{(T)} \right|. \tag{5.5}$$

For all experiments, I also report the AE I would obtain from using the observed label proportions in the training sample as a prediction (labeled "Train"). Although this does not correspond to an interesting prediction (as it only says the future will always look exactly like the past), it does represent a fundamental baseline. If a method is unable to do better than this, it suggests that the method has too much measurement error to be useful.

To test for statistically significant differences between methods, I use an omnibus application of the Wilcoxon signed-rank test to compare one method against all others, including a Bonferroni correction for the total number of tests per hypothesis. With 4 datasets, each with 2 sample sizes, comparing against 6 other methods this results in a significance threshold of approximately 0.001.

Finally, in order to connect this work with past literature on estimating proportions, I also include a side experiment with one intrinsically-labeled dataset where I have artificially modified the label proportions in the target corpus by dropping positive or negatively-labeled instances in order to simulate a large prior probability shift between the source and target domains.

5.4.1 Datasets

Here I briefly describe the datasets I have used in this chapter. Note that although this work is primarily focused on applications in which the amount of human-annotated data is likely to be small (which is typical in most social science applications), fair evaluation of these methods requires datasets that are large enough that one can approximate the expected label proportion in the target corpus using the available labels; as such, the following datasets were chosen so as to have a representative sample of sufficiently large intrinsically and extrinsically-labeled data, where documents were time-stamped, with label proportions that differ between time periods.

Media Frames Corpus (MFC): As a primary example of extrinsic labels, I use the annotated data for three issues (immigration, smoking, and same-sex marriage) from the Media Frames Corpus described in §2.4. I treat annotations as indicating the presence or absence of each dimension in the document, and consider each one as a separate subtask. To create a source and target corpus for each subtask, I partition by time, dividing the articles into those published before and after January 1, 2009. Particularly for this dataset, it seems reasonable to posit that the annotation function was relatively constant between source and target, as the annotators worked without explicit knowledge of the article's date.

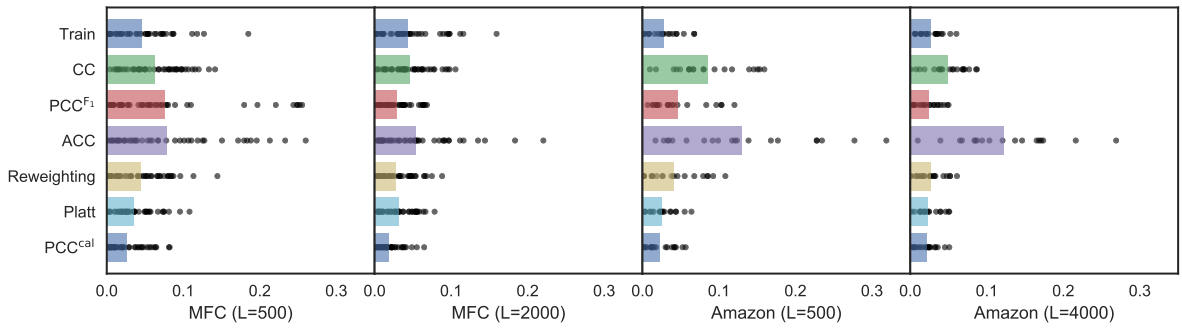
Amazon reviews: As a secondary example of extrinsic labels, I make use of a subset of Amazon reviews for five different product categories, each of which has tens of thousands of reviews. For this dataset, I ignore the star rating associated with the review, and instead focus on predicting the proportion of people that would rate the review as *helpful*. Here I create separate subtasks for each product category by considering each pair of adjacent years as a source and target corpus, respectively (McAuley et al., 2015).

Yelp reviews: As a primary example of a large dataset with intrinsic labels, I make use of the Yelp10 dataset, treating the *location* of the review as the label of interest. Specifically, I create binary classification tasks by choosing three pairs of cities with approximately the same number of reviews, and again use year of publication to divide the data into source and target corpora, creating multiple subtasks per pair of cities. For this experiment, I ignore the star rating, title, and author information, and only consider the review text and location (as a label).

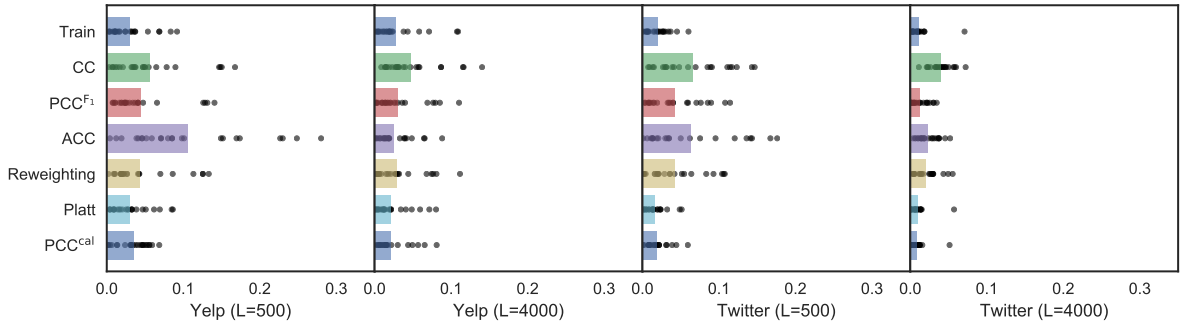
Twitter sentiment: Finally, I include a Twitter sentiment analysis dataset which was collected and automatically labeled, using the presence of certain emoticons as implicit labels indicating positive or negative sentiment (with the emoticons then removed from the text). Because of the way this data was collected, and the relatively narrow time coverage, it seems plausible to treat the sentiment as an intrinsic label. As with the above datasets, I create subtasks by considering all pairs of temporally adjacent days with sufficient tweets, and treating them as a paired source and target corpora, respectively (Go et al., 2009).

5.4.2 Results

The results on the datasets with extrinsic and intrinsic labels are presented in Figures 5.1a and 5.1b, respectively. As expected, the results differ in important ways between intrinsically and extrinsically labeled datasets, although there are some results which hold in all cases. In all settings, CC is worse on average than predicting the observed proportions in the training data (significantly worse for the Amazon and Twitter datasets), reinforcing the idea that averaging the predictions from a classifier will lead to a biased estimate of label proportions. This same finding holds for PCC^{F_1} when the amount of labeled data is small ($L = 500$), suggesting that simply averaging the predicted



(a) Absolute error (AE) on datasets with extrinsic labels. PCC^{cal} (bottom row) performs best on average in all cases and is significantly better than most other methods on MFC.



(b) Absolute error (AE) on datasets with intrinsic labels. No method is significantly better than all others.

Figure 5.1: Performance of all methods on dataset with extrinsic labels (top) and intrinsic labels (bottom). Each dot represents the result for a single subtask, and bars show the mean.

probabilities is not reliable without a sufficiently large labeled dataset.

For the datasets with *extrinsic* labels, PCC^{cal} performs best on average in all settings. For the MFC dataset, PCC^{cal} is significantly better than all methods except Platt scaling when $L = 500$ and significantly better than all methods except reweighting and PCC^{F_1} when $L = 2000$ (after a Bonferroni correction, as in all cases). As expected, ACC is actually worse on average than CC on the extrinsic datasets, presumably because of the mismatched assumptions. Reweighting for covariate shift offers mediocre performance in all settings, perhaps because, while it attempts to account for covariate shift, it may still suffer from miscalibration.

On the datasets with *intrinsic* labels, by contrast, no one method dominates the

others. As expected, ACC does poorly when the amount of labeled data is small ($L = 500$); it does improve upon CC when $L = 4000$, but not by enough to do significantly better than other methods, perhaps calling into question the validity of the assumption that $p(\boldsymbol{x} \mid y)$ is constant in these datasets.

Surprisingly, both Platt scaling and PCC^{cal} also offer competitive performance in the experiments with intrinsic labels. However, this is likely the case in part because the change in label proportions is relatively small from year to year (or day to day in the case of Twitter). This is illustrated by Figure 5.2, which presents the results of the side-experiment with artificially modified (intrinsic) label proportions using a subset of the Twitter data. These results confirm past findings, and show that ACC drastically outperforms other methods such as PCC^{F_1}, *if* one selectively drops instances so as to enforce a large difference in label proportions between source and target. This is the expected result, as ACC is the only method tailored to deal with prior probability shift (which is being artificially simulated). Unfortunately, its advantage is not maintained when the difference between source and target is small, which is the case for all of the naturally-occurring differences found in the Yelp and Twitter datasets. Although past work has relied heavily on these sorts of simulated differences and artificial experiments, it is unclear whether they are a good substitute for real-world data, given that I mostly observed relatively small differences in practice.

Finally, I also tested the effect of using l_2 instead of l_1 regularization, but found that it tended to produce significantly worse estimates of proportions using CC and PCC^{F_1} on the datasets with extrinsic labels, and statistically indistinguishable results using other methods, suggesting that either type of regularization could serve as a basis for PCC^{cal} or Platt scaling.

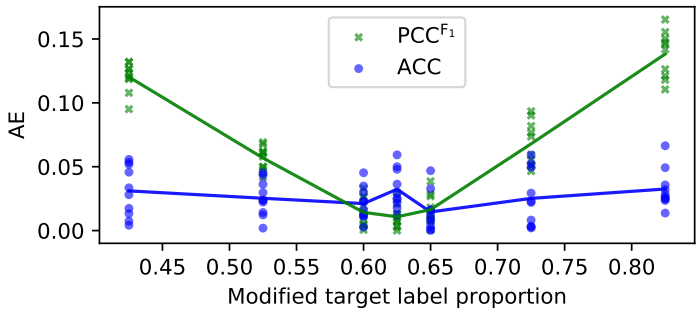


Figure 5.2: Absolute error (AE) for predictions on one day of Twitter data ($L = 5000$) when artificially modifying target proportions. The proportion of positive labels in the source corpus is 0.625. ACC performs significantly better given an large artificially-created difference in label proportions between source and target, but not when the difference is small.

5.5 Discussion

As anyone who has worked with human annotations can attest, the process of collecting annotations is messy and time-consuming, and tends to involve large numbers of disagreements (Artstein and Poesio, 2008). Although it is conventional to treat disagreements as *errors* on the behalf of some subset of annotators, this chapter provides an alternative way of understanding these. By treating annotation as a stochastic process, conditional on text, one can explain not only the disagreements between annotators, but also the lack of self-consistency that is also sometimes observed. Although the assumption that $p(y \mid \boldsymbol{x})$ does not change is clearly a simplification, it seems reasonable when working with trained annotators. Certainly this assumption seems much better justified than the conventional assumption that $p(\boldsymbol{x} \mid y)$ is constant, since the latter does not account for differences in the distribution of text arising from differences in subject matter, etc.

Although I have demonstrated that using a method that is appropriate to the data generating process is beneficial, it is important to note that all methods presented here can still result in relatively large errors in the worst cases. In part this is due to the

difficulty of learning a conditional distribution involving high-dimensional data (such as text) with only a limited number of annotations. Even with much more annotated data, however, previously unseen features could still have a potentially large impact on future annotations. Ultimately, one should be cautious about all such predictions, and always validate where possible, by eventually sampling and annotating data from the target corpus.

What if one can sample from the target corpus? Although there are many situations in which domain adaptation is unavoidable (such as predicting public opinion from Twitter in real time with models trained on the past), at least some research projects in the humanities and social sciences might reasonably have access to all data of interest from the beginning of the project, such as when working with a historical corpus. Although a full proof is beyond the scope of this chapter, in this case, the best approach is almost certainly to simply sample a random set of documents, label them using the annotation function, and report the relative prevalence of each label (Hopkins and King, 2010).

Although this *simple random sampling* (SRS) approach ignores the text, it is an unbiased estimator with variance that can easily be calculated, at least in approximation.[9] More importantly, because it is independent of the dimensionality of the data, it works well on high-dimensional data, such as text, whereas classification-based approaches will struggle.

I illustrate this by comparing SRS and PCC using a simple simulation. Figure 5.3 shows the mean AE for a case in which one knows the true model and only need to

[9] If one were sampling with replacement, the variance in the binary case would be given by the standard formula $\mathbb{V}[\hat{q}^{\text{SRS}}] = \frac{\bar{p}(1-\bar{p})}{L}$, where $\bar{p} = \frac{1}{N_T}\sum_{i=1}^{N_T} p(y_i = 1 \mid \boldsymbol{x}_i)$. This may not be possible, however, as annotators seeing a document for the second or third time would likely be affected by their own past decisions. Nevertheless, using this as the basis for a plug-in estimator should still be a reasonable approximation when the target corpus is large.

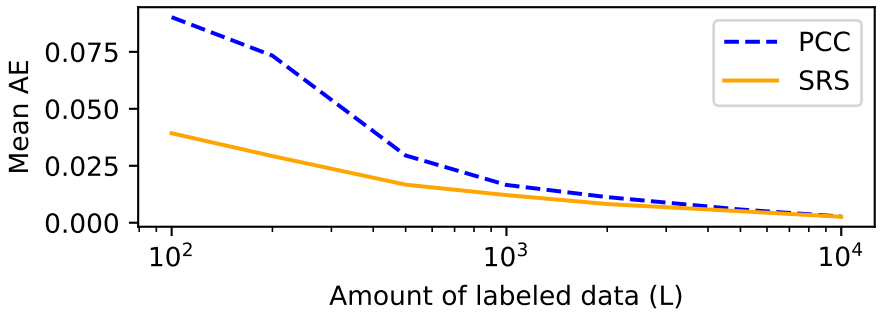


Figure 5.3: Comparison of SRS and PCC in simulation when one knows the true model and sample from the target corpus (averaged over 200 repetitions).

learn the values of the weights. Specifically, I use the following data generating process, for $i = 1, \ldots, N$ and $j = 1, \ldots, P$:

$$
\begin{aligned}
X_{ij} &\sim \text{Bernoulli}(p_x) \\
\beta_j &\sim \text{Laplace}(0, 1) \\
\beta_0 &\sim \mathcal{N}(0, 1) \\
p_i &= \text{sigmoid}(X_{i,:} \cdot \beta + \beta_0) \\
y_i &\sim \text{Bernoulli}(p_i),
\end{aligned}
$$

with $N = 20000$, $P = 10000$, and $p_x = 0.01$, averaged over 200 repetitions. I then fit this model to a subset of the data using an l_1-regularized logistic regression model with regularization strength equal to 1, and average the predicted probabilities over all instances (PCC), or simply average the observed labels in the subset (SRS). Even in this idealized scenario, SRS remains better than PCC for all values of L.

Thus, depending on the level of accuracy required, simply sampling a few hundred documents and labeling them should be sufficient to get a reasonably reliable estimate of the overall label proportions, along with an approximate confidence interval. Unfortunately, this option is only available when one has full access to the target corpus at

the time of annotation.

Additional related work. There is a small literature on the problem of estimating proportions in a target dataset (see §5.1); as I have emphasized, almost all of it makes the assumption that $p(\boldsymbol{x} \mid y)$ is the same for both source and target. Moreover, most of the methods that have been proposed have been tested using relatively small datasets, or datasets where the target corpus has been artificially modified by altering the label proportions in the target corpus (as I did in the side experiment reported in Figure 5.2). It is not obvious that this is a good simulation of the kind of shift in distribution that one is likely to encounter in practice.

An exception to this is Esuli and Sebastiani (2015), who test their method on the RCV1-v2 corpus, also splitting by time. They perform a large number of experiments, but unfortunately, nearly all of their experiments involve only a very small difference in label proportions between the source and target (with the vast majority < 0.01), which limits the generalizability of their findings. Additional methods for calibration could also be considered, such as the isotonic regression approach of Zadrozny and Elkan (2002), but in practice one would expect the results to be very similar to Platt scaling. More recently, Keith and O'Connor (2018) approached this problem by assuming intrinsic labels and fitting simple graphical models to estimate label proportions.

Another line of work has approached the problem of aggregating labels from multiple annotators (Raykar et al., 2009; Hovy et al., 2013; Yan et al., 2013). That is, if one believes that some annotators are more reliable than others, it might make sense to try to determine this in an unsupervised manner, and give more weight to the annotations from the reliable annotators. This seems particularly appropriate when dealing with uncooperative annotators, as might be encountered, for example, in crowdsourcing (Snow et al., 2008; Zhang et al., 2016). However, with a team of trained annotators, it is

likely that legitimate disagreements contain valuable information better not ignored.

Finally, this work also relates to the problem of *active learning*, where the goal is to interactively choose instances to be labeled, in a way that maximizes accuracy while minimizing the total cost of annotation (Beygelzimer et al., 2009; Baldridge and Osborne, 2004; Rai et al., 2010; Settles, 2012). This is an interesting area that might be productively combined with the ideas in this chapter. In general, however, the use of active learning involves additional logistical complications and does not always work better than random sampling in practice (Attenberg and Provost, 2011).

5.6 Summary

When estimating proportions in a target corpus, it is important to take seriously the data generating process. In this chapter, I have argued that in the case of data annotated by humans in terms of categories designed to help answer social-scientific research questions, labels should be treated as *extrinsic*, generated probabilistically conditional on text, rather than as a combination of correct and incorrect judgements about a label *intrinsic* to the document. Moreover, it is reasonable to assume in this case that $p(y \mid \boldsymbol{x})$ is unchanging between source and target, and methods that aim to learn a well-calibrated classifier, such as PCC$^{\text{cal}}$, are likely to perform best. By contrast, if $p(\boldsymbol{x} \mid y)$ is unchanging between source and target, then various correction methods from the literature on estimating proportions, such as ACC, can perform well, especially when differences are large. Ultimately, any of these methods can still result in large errors in the worst cases. As such, validation remains important when treating the estimation of proportions as a type of measurement.

Chapter 6

Transparent and credible predictions using deep neural networks

(This chapter was originally published as Card et al., 2019)

6.1 Introduction

As discussed in the previous chapter, text classification is a useful tool in many social science investigations. Over the past several years, deep learning has become the dominant approach to training machine learning classifiers for any domain involving complex, structured, or high-dimensional data. However, despite the success of deep learning as a framework, many concerns have been raised about deep models. In addition to typically requiring large amounts of labeled data and computational resources, the parameters tend to be relatively difficult to interpret, compared to more traditional (e.g., linear) methods. Moreover, some deep models tend to be poorly calibrated relative to simpler models, despite being more accurate (Guo et al., 2017), and extensive work on adversarial examples has demonstrated that many deep models are more brittle

than test-set accuracies would suggest (Goodfellow et al., 2015; Hendrycks and Gimpel, 2017; Nguyen et al., 2015a; Recht et al., 2018). All of these issues raise the question of whether deep learning is appropriate for social science applications.

Although deep learning is somewhat vaguely defined, I will use it here to refer to any architecture which makes a prediction based on the output of a function involving a series of linear and non-linear transformations of the input representation. While the details of these transformations differ by domain (for example, two-dimensional convolutions are often used for images, whereas sequential models with attention are more common for text), most models for binary or multiclass classification include a final softmax layer to produce a properly normalized probability distribution over the label space. In this chapter, I explore an alternative to the softmax, yielding what I call a *deep weighted averaging classifier* (DWAC), and evaluate its potential to deliver equally accurate predictions, while offering greater transparency, interpretability, and robustness.

Particularly in light of recent controversy and legislation, such as the General Data Protection Regulation (GDPR) in Europe, there has been rapidly growing interest in developing more interpretable models, and in finding ways to provide explanations for predictions made by machine learning systems. Although there is currently an active debate in the field about how best to conceptualize and operationalize these terms (Doshi-Velez and Kim, 2017; Guidotti et al., 2018), recent research has broadly fallen into two camps. Some work has focused on models that are inherently interpretable, such that an explanation for a decision can be given in terms that are easily understood by humans. This category includes classic models that can easily be simulated by humans, such as decision lists, as well as sparse linear models, where the prediction is based on a weighted sum of features (Breiman et al., 1984; Lakkaraju et al., 2016; Lou et al., 2012; Tibshirani, 1996; Ustun and Rudin, 2016; Wang and Rudin, 2015). Other

work, meanwhile, has focused on developing methods to provide explanations that approximate the true inner workings of more complex models, in a way that provides some utility to the user or developer of a model beyond what is attainable through more direct means (Bastani et al., 2017; Lei et al., 2016; Lundberg and Lee, 2017; Ribeiro et al., 2016, 2018b,a; Selvaraju et al., 2016).

In this chapter, I propose a method which, like those in the former category, offers an explanation that is transparent (in that the complete explanation is in terms of a weighted sum of training instances), but also explore ways to approximate this explanation by using only a subset of the relevant instances. While this approach retains some of the inherent complexity of typical deep models (in that it is still difficult to explain *why* the model has weighted the training instances as it has for a particular test instance), the mechanism behind the prediction is far more transparent than softmax-based models, and the individual instance weights provide a way for a user to examine the basis of the prediction, and evaluate whether or not the model is doing something reasonable. Similarly, while looking at the nearest neighbors of a test point is a commonly-used heuristic to attempt to understand what a model is doing, that approach is only an approximation for models which map each instance directly to a vector of probabilities.

There is, of course, a long history in machine learning of making predictions directly in terms of training instances, including nearest neighbor methods (Cover and Hart, 1967), kernel methods, including support vector machines (Boser et al., 1992; Cortes and Vapnik, 1995), and transductive learners more broadly (Vapnik, 1998). The main novelty here is to adapt any existing deep model to make predictions explicitly in terms of the training data using only a minor modification to the model architecture, and arguing for and demonstrating the advantages offered by this approach.

As I will describe in more detail below, I propose to learn a function which maps

from the input representation to a low-dimensional vector representation of each input. Predictions on new instances are then made in terms of a weighting of the label vectors of the training instances, where the weights are a function of the distance from the instance to be evaluated to all training instances, in the low-dimensional space. This is closely related to a long line of past work on *metric learning* (Xing et al., 2002; Goldberger et al., 2004; Weinberger et al., 2006; Davis et al., 2007; Bellet et al., 2013; Kulis, 2013), but rather than trying to optimize a particular notion of distance (such as Mahalanobis distance), I make use of a fixed distance function, and allow the architecture of standard deep models to do the equivalent work. This idea is also related to models which use neural networks to learn a similarity function for specific applications, such as face recognition (Chopra et al., 2005) or text similarity (Mueller and Thyagarajan, 2016), and similar architectures have also been used for one-shot learning (Koch et al., 2015; Vinyals et al., 2016); here I show how this is a more generally applicable way to train models, and I emphasize the connection to interpretability.

Such an approach comes with distinct advantages:

1. A precise explanation of why the model makes a specific prediction (label or probability) can be given in terms of a subset of the training examples, rank ordered by distance. Moreover, the weight on each training instance implicitly captures the degree to which the model views the two instances as similar. The explanation is thus given in terms of exemplars or *prototypes*, which have been shown to be an effective approach to interpretability (Kim et al., 2014).
2. In §6.5.3, I show that, in many cases, a very small subset of training instances can be used to provide an approximate explanation with high fidelity to the complete explanation.
3. In addition, it is possible to choose the size of the learned output representation so as to trade off between performance and interpretability. For example, one can

use a lower dimensional output representation if one wishes to make it easy to directly visualize the embedded training data.

4. Even in cases where revealing the training data is not feasible, it is possible to provide an explanation purely in terms of weights and labels. Although this does not reveal the *way* in which a new instance is viewed (by the model) as similar to past examples, it still provides a quantifiable notion of how unusual the new example is. The form of this model suggests a natural metric of nonconformity, and in §6.3.4, I formalize this using the notion of *conformal methods*, describing how the relevant distances can be used to either provide bounds on the error rate (for data drawn from the same distribution), or robustness against outliers.

5. Finally, although this model does entail a slight cost in terms of increased computational complexity, the difference in terms of speed and memory requirements at test time can be minimized by pre-computing and storing only the low-dimensional representation of the training data (from the final layer of the model). The cost during training will in most cases be dominated by the other parts of the network, and it is still possible to train such models on large datasets without difficulty. Moreover, in the experiments, this choice seemingly involves no loss in accuracy or calibration.

Deep weighted averaging classifiers (DWACs) are ideally suited to domains where it is possible to directly inspect the training data, such as controlled settings like social science research. In this domain, DWACs offer a more transparent and interpretable version of any successfully developed deep learning architecture. Although the advantages are diminished in domains where privacy is a concern, presenting information solely in terms of weights and labels still provides a useful way to quantify the credibility of a prediction, even without allowing direct inspection of the original training data.

The experiments in this chapter are primarily based on benchmark datasets in order

to provide a robust comparison to the baseline. However, DWACs provide a compelling alternative to the use of conventional deep learning models in socially consequential applications, especially when developers wish to maximize accuracy while retaining the ability to easily investigate the reasons for individual predictions. Moreover, because conformal methods provide guarantees on error rates, albeit in a somewhat unusual fashion, this suggests potential value in terms of being able to quantify uncertainty in measurement.

6.2 Background

6.2.1 Scope and notation

In this chapter, I will be concerned with the problem of classification. In general, I will assume a set of m training instances, $\boldsymbol{x}_i$ for $i = 1, \ldots, m$, with corresponding labels in some categorical label space, $y_i \in \mathcal{Y}$, for $i = 1, \ldots, m$, where $c = |\mathcal{Y}|$ is the number of classes. I also assume that there will eventually be given a set of n test instances, $\boldsymbol{x}_i, y_i$, for $i = 1 + m, \ldots, n + m$. I will use $\boldsymbol{h}_i$ to refer to the output representation of a model for instance i. Square brackets with a subscript $[\cdot]_k$ will denote the kth element of a vector.

6.2.2 Nonparametric kernel regression

DWACs build on a classic method from nonparametric regression, known as Nadaraya-Watson (NW). The original use case of NW was in regression, with $y_i \in \mathbb{R}$, where it is assumed that $y_i = m(\boldsymbol{x}_i) + \varepsilon$, where $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{V}[\varepsilon] = \sigma^2$. The goal is to estimate the

mean function, $m(\boldsymbol{x})$, which can be expressed in terms of the joint density as

$$m(\boldsymbol{x}) = \mathbb{E}[Y \mid X = \boldsymbol{x}] = \int y\, P(y \mid \boldsymbol{x})\, dy = \frac{\int y\, P(\boldsymbol{x}, y)\, dy}{\int P(\boldsymbol{x}, y)\, dy}. \tag{6.1}$$

By approximating the joint density using kernel density estimation, it is possible to re-express this as a weighted sum of training instances, i.e.,

$$\hat{m}(\boldsymbol{x}) = \frac{\sum_{i=1}^{m} y_i\, K(\boldsymbol{x}, \boldsymbol{x}_i)}{\sum_{j=1}^{m} K(\boldsymbol{x}, \boldsymbol{x}_j)}, \tag{6.2}$$

where $K(\boldsymbol{x}, \boldsymbol{x}_i)$ is a kernel, such as a Gaussian (Nadaraya, 1964; Watson, 1964). It is easy to see that this corresponds to a *linear smoother*, in that it will predict outputs as a weighted sum of training instances, i.e.,

$$\hat{m}(\boldsymbol{x}) = \sum_{i=1}^{m} y_i\, \alpha_i(\boldsymbol{x}), \tag{6.3}$$

where $\alpha_i(\boldsymbol{x}) = \frac{K(\boldsymbol{x}, \boldsymbol{x}_i)}{\sum_{j=1}^{m} K(\boldsymbol{x}, \boldsymbol{x}_j)}$.

This method can easily be adapted to classification by predicting the *probability* of an output label as a weighted sum of training labels, i.e.,

$$P_{\text{NW}}(y = k \mid \boldsymbol{x}) = \sum_{i=1}^{m} \mathbb{I}[y_i = k]\, \alpha_i(\boldsymbol{x}), \tag{6.4}$$

where $\mathbb{I}[\cdot]$ equals 1 if the condition holds or 0 if not.

The primary limitation on NW is due to the curse of dimensionality: as the dimensionality of $\boldsymbol{x}$ grows, sparse data becomes a problem, and the notion of proximity becomes problematic (Aggarwal et al., 2001). In this work, I show how one can avoid this problem by using neural networks to embed inputs into a space with much lower dimensionality, and proceed to compute weights over training instances in that space.

6.2.3 Conformal methods

Conformal methods refer to a broad set of ideas which aim to provide theoretical guarantees on error rates in classification or regression (Saunders et al., 1999; Vovk et al., 2005; Shafer and Vovk, 2008; Lei et al., 2014). Conformal methods can be used with any base classifier or predictor, and work by introducing a generic notion of *nonconformity*. As will be explained in detail below, each possible prediction (i.e., label or value) that can be made for a given test instance can be evaluated in terms of its nonconformity. By comparing these with the nonconformity scores of either all data in a leave-one-out manner, or with a held-out *calibration* set, the equivalent of a p-value can be associated with each possible prediction, allowing for thresholding in a way that provides a guarantee on the error rate (for i.i.d. or exchangeable data).

Because of the high computational cost of the leave-one-out approach to conformal predictors, I will focus here on the approach based on a held-out calibration set.[1] In particular, I will begin by shuffling the training data, and partitioning it into a proper training set ($i = 1, \ldots, t$), and a calibration set ($i = t+1, \ldots, m$).[2]

The fundamental choices in conformal methods are a base classifier and a measure of nonconformity, A. The latter concept, which intuitively corresponds to how *atypical* an instance is, maps a bag of examples (i.e., the proper training set, with labels), and one additional instance ($\boldsymbol{x}$, the observed features) with one possible label, $k \in \mathcal{Y}$, to a scalar $\eta \in \mathbb{R}$, i.e.,

$$\eta(\boldsymbol{x}, k) = A\left(\{\!\{(\boldsymbol{x}_i, y_i)\}\!\}_{i=1}^{t}, (\boldsymbol{x}, k)\right), \tag{6.5}$$

where $\{\!\{\cdot\}\!\}$ denotes a *bag*, i.e., a multi-set (potentially containing duplicate instances).

[1] The leave-one-out approach may be superior in terms of statistical efficiency, but it is computationally infeasible for all but very small datasets.

[2] The calibration set will also serve as the validation set for early stopping during training.

The idea of a measure of nonconformity is quite general, but in practice, the most common approach is to convert a bag of examples into a *model*, and then compare the *prediction* of that model on the training instance ($\boldsymbol{x}$) with the hypothesized label, k. In particular, for any probabilistic model, the simplest measure of model-based nonconformity is any inverse monotonic transform of the predicted probability of the hypothesized label, such as the inverse or negation. For example, the following is a valid measure of nonconformity:

$$\eta(\boldsymbol{x}, k) = -P_{\mathcal{M}\left(\langle(\boldsymbol{x}_i, y_i)\rangle_{i=1}^{t}\right)}\left(y = k \mid \boldsymbol{x}\right), \tag{6.6}$$

where $\mathcal{M}\left(\langle(\boldsymbol{x}_i, y_i)\rangle_{i=1}^{t}\right)$ represents a model trained on the proper training set, $i = 1, \ldots, t$.

Conformal methods work by comparing the nonconformity score of each possible prediction for each test instance to the nonconformity scores of all instances in the calibration set. Specifically, the model will compute a p-value for each hypothetical test instance label equal to the proportion of the calibration instances that have a higher nonconformity score (i.e., a value between 0 and 1 indicating how *conforming* a possible label for a new instance is, relative to the calibration data). More precisely, for a test instance $\boldsymbol{x}$ and hypothesized label, k,

$$p(\boldsymbol{x}, k) = \frac{\sum_{j=t+1}^{m} \mathbb{I}[\eta(\boldsymbol{x}_j, y_j) \geq \eta(\boldsymbol{x}, k)]}{m - t}, \tag{6.7}$$

where η is computed relative to the proper training set for all instances (both calibration and test).

Unlike traditional classifiers, conformal methods may predict anywhere from zero to c labels for a given instance. I will revisit this choice below, but for the moment, the decision rule will be that for each test instance, the model will make a positive

prediction for all labels k for which $p(\boldsymbol{x}, k) > \varepsilon$, where $\varepsilon \in [0, 1]$ is chosen by the user. By the properties of conformal predictors, these predictions will be asymptotically valid; that is, both theoretically and empirically, for i.i.d. data, this will produce a set of predicted labels for each test instance such that at least $1 - \varepsilon$ of the predicted label sets include the true label, with high probability. Moreover, this property holds for any measure of nonconformity.[3] Of course, this property is trivially easy to satisfy by predicting all labels for all test instances. In practice, however, better choices of measures of nonconformity will be more or less *efficient*, in that they will tend to produce smaller predicted label sets without compromising on error rate.

Although there is some risk of terminological confusion here, Saunders et al. (1999) propose to characterize the distribution of p-values for a single instance in terms of what they call *confidence* and *credibility* (see also Shafer and Vovk, 2008). In the context of conformal methods, "confidence" refers to the largest $1-\varepsilon$ such that the predicted label set includes only a single label (i.e., one minus the second-largest p-value among the possible labels). This corresponds to the probability, according to the model, that the label which is predicted to be most likely is correct. For example, for i.i.d. data, one would expect 92% of predictions with 92% confidence to be correct, and 8% to be incorrect.

"Credibility", by contrast, is equal to the largest ε for which the predicted label set is empty (i.e., the largest p-value among the possible labels). This correspond to one minus the model's confidence that *none* of the possible labels are correct. In other words, predictions with low credibility indicate that even the most likely prediction is relatively nonconforming compared to the calibration instances (with their true labels),

[3] More precisely, for any measure of nonconformity, and i.i.d. data drawn from any distribution, the unconditional probability that the true label of a random test instance is not in the predicted label set does not exceed ε for any ε. The broader statement follows from the law of large numbers. It is also possible to bound the conditional probability of error (conditional on the training data) with slightly weaker guarantees. For more details, please refer to Vovk et al. (2005) and Vovk (2012).

and that one should therefore be skeptical of this prediction.

Terminology aside, the important point is that it is these two properties in combination which fully characterize how to interpret a prediction. In particular, when a prediction has low credibility (which also entails low confidence), this implies that none of the labels are suitable for the corresponding instance, according to the model. By contrast, if multiple p-values are close to one, (high credibility, but low confidence), this corresponds to what is sometimes called *ambiguity*, that is, multiple labels seem suitable, according to the model. Only when there is a single label that has a high p-value (high confidence and high credibility), do we have the desired scenario in which the model will predict a single label.

The idea of not predicting any label is somewhat foreign to conventional approaches, as one can only obtain better accuracy by hazarding at least some guess, and in some applications it would be reasonable to still predict the most probable label according to the model, with an associated confidence and credibility. In other circumstances, however, there may be valuable information in an empty label set. In particular, as I will discuss below, an empty label set (low credibility) is an indication that none of the labels are appropriate, suggesting that the model itself may be inappropriate for that particular instance (or equivalently, that the instance may be out-of-domain for that model).

6.3 Deep weighted averaging classifiers

I now turn to the method proposed in this chapter, deep weighted averaging classifiers.

6.3.1 Model details

Most neural network models for classification take the form

$$P(y = k \mid \boldsymbol{x}) = \frac{\exp([\boldsymbol{h}]_k)}{\sum_{j=1}^{c} \exp([\boldsymbol{h}]_j)}, \tag{6.8}$$

where $\boldsymbol{h} = \boldsymbol{W} \cdot f(\boldsymbol{x}) + \boldsymbol{b}$. In this architecture, $f(\boldsymbol{x})$ embeds the input in a model-specific way (e.g., a convolutional or recurrent network) into a lower-dimensional vector representation. Although $\boldsymbol{W}$ and $\boldsymbol{b}$ could be folded into $f(\boldsymbol{x})$, I make them explicit to emphasize that $\boldsymbol{W}$ is required to project the output of $f(\boldsymbol{x})$ (which is a vector of arbitrary length) down to a vector of length c (the number of classes). The softmax (Eq. 6.8) then projects this vector onto the simplex.

My proposed alternative is to leave $f(\boldsymbol{x})$ unchanged, eliminate the softmax, and to redefine the predicted probability as

$$P(y = k \mid \boldsymbol{x}) = \frac{\sum_{i=1}^{t} \mathbb{I}[y_i = k]\; w(\boldsymbol{h}, \boldsymbol{h}_i)}{\sum_{j=1}^{t} w(\boldsymbol{h}, \boldsymbol{h}_j)}, \tag{6.9}$$

i.e., a weighted average of the labels of the instances in the proper training set, where $\boldsymbol{h}$ is defined as above and $w(\boldsymbol{h}, \boldsymbol{h}_i)$ is a function of the similarity between the embeddings of $\boldsymbol{x}$ and $\boldsymbol{x}_i$ in the low-dimensional space, according to some metric. In this architecture, the dimensionality of $\boldsymbol{h}$ is arbitrary, and the size of $\boldsymbol{W}$ and $\boldsymbol{b}$ can be modified as necessary.[4]

An obvious choice of weight function is a Gaussian kernel operating on Euclidean distance, i.e.,

$$w(\boldsymbol{h}, \boldsymbol{h}_i) = \exp\left(\frac{-\|\boldsymbol{h} - \boldsymbol{h}_i\|_2^2}{2\sigma}\right). \tag{6.10}$$

[4] One could of course dispense with $\boldsymbol{W}$ and $\boldsymbol{b}$ here, and compute $w(\cdot, \cdot)$ directly on the output of $f(\boldsymbol{x})$, but I wish to remain as close as possible to the softmax model for the purpose of comparison, while allowing for the possibility of varying the size of $\boldsymbol{h}$.

Typically, in using NW, or other kernel smoothers, one needs to choose the bandwidth, equivalent to σ in equation (6.10). However, because I will assume that this classifier will be built on top of a high-capactity embedding network, $f(\boldsymbol{x})$, I will simply fix $\sigma = \frac{1}{2}$, and force $f(\boldsymbol{x})$ to adapt to this distance function.

6.3.2 Training

In order to learn $\boldsymbol{W}$, $\boldsymbol{b}$, and all parameters of $f(\boldsymbol{x})$, I will use stochastic gradient descent to optimize the log loss on the training data. Because it is impractical to compute the exact probabilities according to the model during training (because they depend on all training instances), I instead rely on an approximation based on the other instances within each minibatch. Specifically, on each epoch of training, I shuffle all instances in the proper training set into minibatches of size B. For each minibatch, $\mathcal{B}$, I then minimize

$$\mathcal{L}(\mathcal{B}) = \frac{1}{B} \sum_{j \in \mathcal{B}} \sum_{k=1}^{c} -\mathbb{I}[y_j = k] \log \hat{P}(y_j = k \mid \boldsymbol{x}_j), \tag{6.11}$$

where

$$\hat{P}(y_j = k \mid \boldsymbol{x}_j) = \frac{\sum_{i \in \mathcal{B} \setminus \{j\}} \mathbb{I}[y_i = k] \; w(\boldsymbol{h}_j, \boldsymbol{h}_i)}{\sum_{l \in \mathcal{B} \setminus \{j\}} w(\boldsymbol{h}_j, \boldsymbol{h}_l)}. \tag{6.12}$$

As usual, the loss function in equation (6.11) can be augmented with a regularization term if desired.

Although computing all pairwise distances between many points is relatively expensive, this can be done efficiently for minibatches using standard matrix operations on a GPU. Specifically, a forward pass through the last layer of a DWAC model (i.e., computing probabilities for one minibatch) requires $\mathcal{O}(B^2 h)$, where B is the size of the

minibatch and $h = \text{len}(\boldsymbol{h})$. This will typically be larger than the last layer of the softmax classifier, which is $\mathcal{O}(Bhc)$, where c is the number of classes, but in most cases will be dominated by the cost of computing $f(\boldsymbol{x})$. In practice, the only significant increase in training time is due to the need to embed all training instances in order to estimate performance on a validation set after each epoch (which could also be approximated). As such, the training runtime will tend to be no worse than twice that of training a softmax model. Similarly, at test time, the computational cost of making a prediction on one test instance is dominated by the cost of embedding the training data. However, this can be pre-computed after training, and only the low-dimensional $\boldsymbol{h}$ vectors need be stored.[5]

6.3.3 Prediction and explanations

Once the model has been trained, predictions can easily be made using the entire training dataset, rather than using a subset, as when computing the loss during training. The explanation for why the model predicts a particular label or probability can then be given explicitly in terms of the training instances, along with the weight on each instance. Moreover, if one considers the closest points (which will be most heavily weighted) as being the most similar, relevant, or important, it is reasonable to provide a sorted list of examples as the explanation. Because the later examples will carry less weight, in many cases only a subset of instances needs to be provided (because for many instances, the lower-weighted training instances will be unable to affect the prediction, no matter what their labels may be).

If one wishes to provide an even simpler but approximate explanation, one can also choose to provide only the closest k examples as the explanation, which is a

[5] Note that one only needs to compute distances in the low-dimensional space, for which one can choose an appropriately small dimensionality.

commonly used heuristic for trying to understand model behavior. Although one would not expect that using only a small set of examples would provide a well-calibrated *probability*, it could still provide a reasonable approximate explanation for why the model predicted a particular *label*, assuming that there is strong agreement between predictions made using such a subset and the full model (which I will empirically evaluate in the experiments below).

6.3.4 Confidence and credibility

As discussed in section §6.2.3 above, for any probabilistic classifier, one can use any monotonic transformation of the predicted probabilities which reverses their order (such as $-P(\boldsymbol{x})$ or $1/P(\boldsymbol{x})$) as a valid measure of nonconformity. However, the architecture of the model proposed in this chapter suggests another measure, namely the negated unnormalized weighted sum of training labels of the hypothesized class, i.e.,

$$\eta(\boldsymbol{x}, k) = A\left(\langle(\boldsymbol{x}_i, y_i)\rangle_{i=1}^{t}, (\boldsymbol{x}, k)\right) = -\sum_{i=1}^{t} \mathbb{I}[y_i = k]\, w(\boldsymbol{h}, \boldsymbol{h}_i). \tag{6.13}$$

Because of the properties of conformal methods, this measure of nonconformity is automatically *valid*. It may, however, be more or less *efficient* than other measures, such as ones based on probabilities. I note, however, that this proposed measure has an intuitive explanation in terms of how close the training points of the predicted class are (in the embedded space) to the instance for which one wishes to make a prediction. Naturally the absolute distance has no meaning in the embedded space, but I avoid this problem by scaling the measure of nonconformity relative to calibration data in order to obtain a p-value, as is always the case in conformal methods.

When using probability as the basis of nonconformity, the farther a point is from

the decision boundary, the higher will be its predicted probability, and therefore its credibility. Under the measure I propose in equation (6.13), by contrast, predictions will only be associated with high credibility when the embedded representation of that instance is relatively close to the embedded training instances. That is, if one encounters an instance that is unlike anything seen in the training data, and if the model embeds that instance such that it is far away from all embedded training instances, then this measure will tell us that it is highly nonconforming for all classes, which will result in the model's prediction having very low credibility. As I show below, this is a useful way to quantify the degree to which one should be skeptical of the model's prediction.

6.4 Experiments

To demonstrate the potential of DWAC models, I provide a range of empirical evaluations on a variety of datasets. In addition to showing that my proposed approach is capable of obtaining equivalently accurate predictions, I also compare to a baseline in terms of calibration and robustness to outliers, and illustrate the sorts of explanations offered by a DWAC model. I also empirically validate that the theoretical guarantees claimed by conformal methods hold for both the softmax and the DWAC model, both with and without my proposed measure of nonconformity.

In all cases I report accuracy and calibration (i.e., the accuracy of the predicted probabilities), measuring the latter in terms of mean absolute error (MAE), using the adaptive binning approach of Nguyen and O'Connor (2015). For the initial comparison between models, I only consider conventional prediction (i.e., only using the top-scoring label predicted by each model), and separately evaluate conformal prediction in §6.5.4. Note that the purpose here is not to demonstrate state-of-the-art performance

on any particular task, but rather to show that the proposed modification works for a wide variety of architectures.

In order to evaluate robustness to outliers, I consider two approaches. The first is to drop one class from a multiclass dataset, and treat the held-out class as out-of-domain data. The other approach is to find a dataset that has a similar input representation, but is fundamentally different in terms of content, and again, treat these instances as out-of-domain.

6.4.1 Datasets

For the experiments I make use of datasets of three different types (tabular, image, and text), including both binary and multiclass problems. For tabular data, I use the familiar Adult Income (Kohavi, 1996) and Covertype (Blackard and Dean, 1999) datasets available from the UCI machine learning repository,[6] as well as the Lending Club loan dataset available through Kaggle.[7] For images I use CIFAR-10 (Krizhevsky and Hinton, 2009) and Fashion MNIST (Xiao et al., 2017). For text I use paragraph-length product reviews (Amazon; 1–5 stars) (McAuley et al., 2015) and movie reviews (IMDB; positive or negative) (Maas et al., 2011), sentences extracted from movie reviews and labeled in terms of *subjectivity* (Pang and Lee, 2004), and a dataset of Stack Overflow question titles sampled from 20 different categories (Xu et al., 2015).

Table 6.1 summarizes the most important properties of these datasets. For language datasets, text was tokenized with spaCy,[8] and converted to lower case, using a vocabulary built from the training set, with word embeddings initialized using 300-dimensional Glove vectors trained on the 6 billion tokens of Wikipedia 2014 and Gigaword 5.[9]

[6] https://archive.ics.uci.edu/ml/datasets
[7] https://www.kaggle.com/wsogata/good-or-bad-loan-draft/notebook
[8] https://spacy.io/
[9] https://nlp.stanford.edu/projects/glove/

Dataset	Type	# classes	# instances	# features
Adult Income	Tabular	2	45,000	103
Covertype	Tabular	7	581,000	54
Lending Club	Tabular	2	626,000	157
Fashion MNIST	Image	10	60,000	784
CIFAR-10	Image	10	60,000	3,072
Subjectivity	Text	2	10,000	20,000
Stack Overflow	Text	20	20,000	15,000
IMDB	Text	2	50,000	112,000
Amazon	Text	5	190,000	144,000

Table 6.1: Properties of datasets used in this chapter. For text data, I report vocabulary size as the number of features.

6.4.2 Models and training

In all cases I choose a base model appropriate to the data. For the tabular data, I use a simple three-layer multi-layer perceptron. For images, I use multi-layer convolutional neural networks. For text datasets, I use a shallow convolutional model with attention (Mullenbach et al., 2018). In all cases I compare DWAC and softmax models of equivalent size, but also explore varying the dimensionality of $\boldsymbol{h}$ in the DWAC model. I use the standard train/test split where available, and otherwise sample a random 10% of the data for a test set, and always use a random 10% of the training data as a validation/calibration set. For measuring accuracy and calibration on test data, I average over 5 trials with different splits of the training data into a proper training set and a validation/calibration set, with the same split being given to both models. For both models I use Adam (Kingma and Ba, 2014) with an initial learning rate of 0.001 and early stopping.

	Accuracy ↑	
Dataset	Softmax	DWAC
Adult Income	0.851 (±0.002)	0.850 (±0.002)
Covertype	0.774 (±0.003)	0.760 (±0.001)
Lending Club	0.956 (±0.001)	0.955 (±0.001)
Fashion MNIST	0.928 (±0.002)	0.927 (±0.002)
CIFAR-10	0.898 (±0.004)	0.897 (±0.009)
Subjectivity	0.948 (±0.002)	0.946 (±0.004)
Stack Overflow	0.869 (±0.005)	0.866 (±0.008)
IMDB	0.905 (±0.002)	0.904 (±0.001)
Amazon	0.740 (±0.002)	0.738 (±0.002)

	Calibration (MAE) ↓	
Dataset	Softmax	DWAC
Adult Income	0.012 (±0.006)	0.018 (±0.002)
Covertype	0.005 (±0.001)	0.010 (±0.001)
Lending Club	0.007 (±0.001)	0.014 (±0.001)
FashionMNIST	0.006 (±0.001)	0.003 (±0.001)
CIFAR-10	0.011 (±0.001)	0.009 (±0.002)
Subjectivity	0.020 (±0.006)	0.023 (±0.006)
Stack Overflow	0.009 (±0.001)	0.010 (±0.001)
IMDB	0.029 (±0.006)	0.024 (±0.010)
Amazon	0.008 (±0.002)	0.004 (±0.001)

Table 6.2: Accuracy (top; higher is better) and calibration (bottom; lower is better) on various datasets using the single best-scoring predicted label from softmax and DWAC models of equivalent size, with standard deviations in parentheses.

6.5 Results

6.5.1 Classification performance

As shown in Table 6.2, except for one dataset (Covertype) the performance of DWAC is indistinguishable from a softmax model of the same size in terms of accuracy. As noted above, these results are based on the single best-scoring label from each model, without yet incorporating the idea of conformal prediction. For calibration, the DWAC model is sometimes slightly better and sometimes slightly worse, although I note in passing that, at least for these models and datasets, the predictions from both models

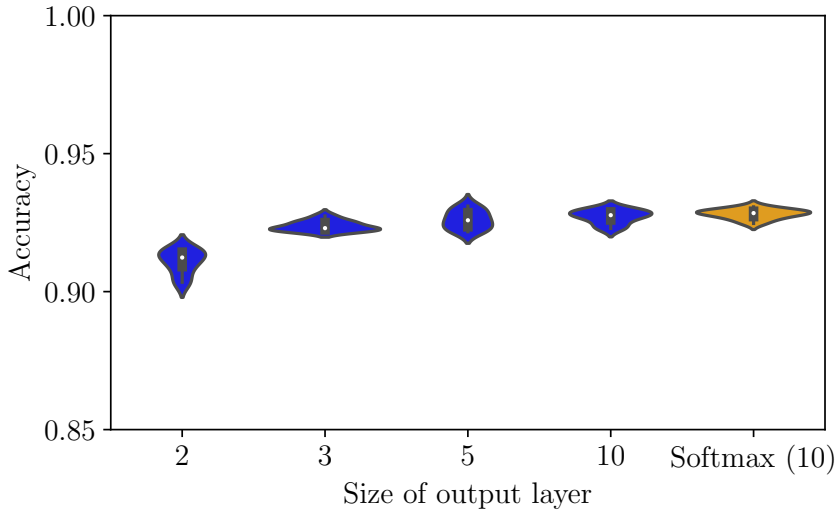


Figure 6.1: Accuracy of DWAC in comparison to a softmax model on the 10-class Fashion MNIST dataset for varying dimensionality of the output layer. Performance is indistinguishable for a DWAC model of the same size, but accuracy drops if the size of the output layer is decreased too much.

are quite well calibrated, such that the predicted probabilities are relatively reliable, at least for in-domain test data. As expected, runtime during training was approximately 50% longer per epoch than for the equivalent softmax model, with a similar number of epochs required.

One advantage of DWAC is the freedom to choose the dimensionality of the final output layer, and Figure 6.1 illustrates the impact of this choice on the performance of the new model. While using the same dimensionality as the softmax model gives equivalent performance, the same accuracy can often be obtained using a lower-dimensional representation (i.e., few total parameters). In some cases, however, reducing the dimensionality too much (e.g., two-dimensional output for Fashion MNIST) results in a slight degradation of performance.

On the other hand, using a two-dimensional output layer means that one are able to more easily visualize the learned embeddings, without requiring an additional dimensionality reduction step, e.g., using PCA or t-SNE (van der Maaten and Hinton, 2008).

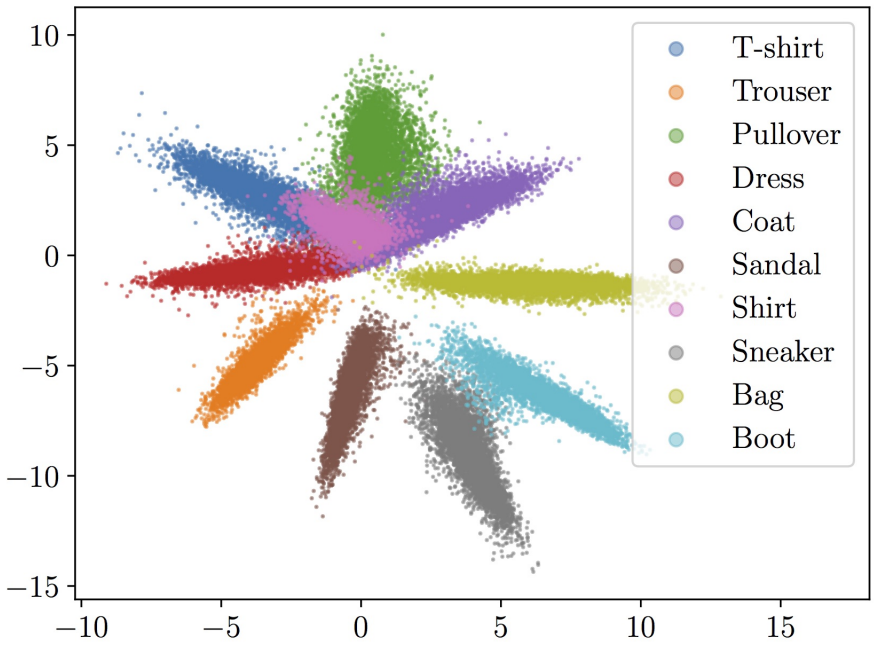


Figure 6.2: Learned embeddings of the Fashion MNIST training data when using a DWAC model with a two-dimensional output layer.

In this way, one could look directly at where a test instance is being embedded, relative to training instances, with no loss of fidelity.

Figure 6.2 shows the embeddings learned by DWAC for the Fashion MNIST training data using a two-dimensional output layer. Pleasingly, there is a natural semantics to this space, with all of the footwear occurring close together, and the "shirt" class being centrally located relative to related classes (t-shirt, pullover, etc.).

6.5.2 Interpretability and explanations

Recall that explanations for predictions made by DWAC are given in terms of a weighted sum of training instances. Figure 6.3 (top) shows an example of a partial explanation provided by DWAC for an image dataset. A test image is shown in the top row, along

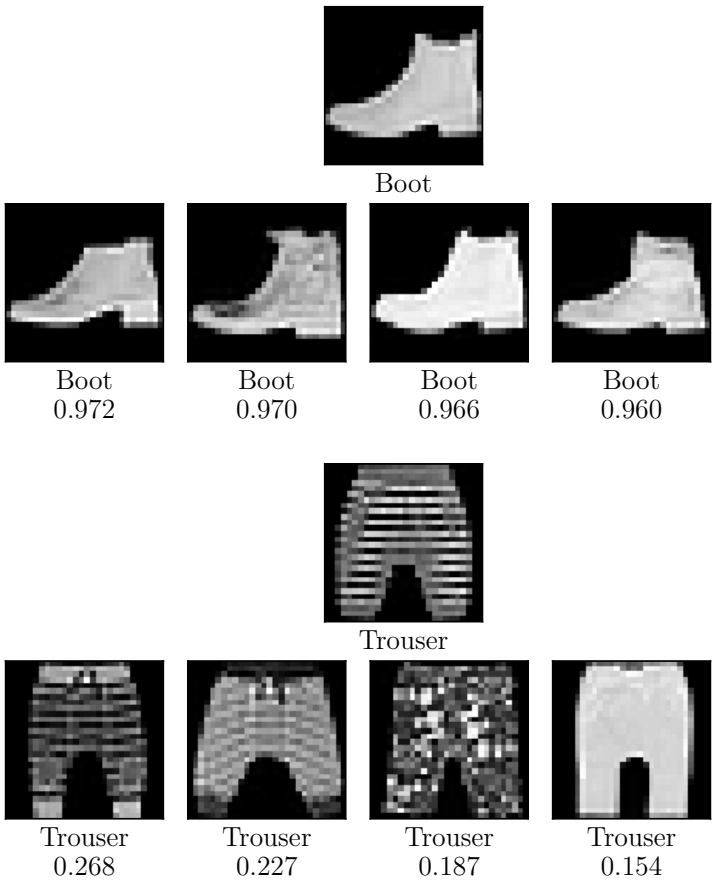


Figure 6.3: Two examples of predictions made by the DWAC model on the Fashion MNIST dataset. An approximate explanation for the model's prediction on the test image (single image) is the four most-highly weighted training images (row of four), along with their weights and labels. The top example shows a prediction that is well supported, with many nearby training instances of the same class, whereas the bottom shows an example where even the nearest training instances are relatively distant.

with its true label, and the four closest images from the proper training set (as measured in the embedded space) are shown below it, along with their weights. In this case, all share the same label, and all contribute approximately equally to the prediction.

As a contrasting example, Figure 6.3 (bottom) shows an example which has poor support among the nearest training points. Although the prediction is correct, and the closest training images appear visually similar, this sort of wide-legged trouser is quite rare in the dataset (most trousers included have narrow legs). As such, the model has

Weight	Sentence	Label
Test	*Drupal 6 dynamic menu item*	*Drupal*
0.994	drupal how to limit the display results of the calendar view	Drupal
0.994	Drupal : Different output for first item in a block	Drupal
0.994	changing user role in drupal	Drupal

Weight	Sentence	Label
Test	*save data from editable division*	*Ajax*
0.214	Pass data from workspace to a function	Matlab
0.133	upload data from excel to access using java	Apache
0.130	Finding incorrectly - formatted email addresses in a CSV file	Excel

Table 6.3: Two examples from the Stack Overflow dataset with approximate explanations from a DWAC model: an easy example with many close neighbours (top) and a more difficult example with no close neighbours (bottom). The first line is the test instance in both cases.

clearly not learned as well how to embed such images into the low-dimensional space. The low weights indicate that one should be skeptical of this prediction.

Images are relatively easy to compare at a glance; text, by contrast, may be more difficult. Nevertheless, the explanations given by DWAC for the predictions on text data are in some cases very meaningful. For the Stack Overflow dataset, for example, many instances are almost trivially easy, in that the label is part of the question. Not surprisingly, these examples tend to have many highly-weighted neighbors which provide a convincing explanation. Such an example is shown in Table 6.3 (top). In other cases, the text is more ambiguous. Table 6.3 (bottom) shows an example with very little support, for which a user might rightly be skeptical of the model, based on both the weights and the explanation.

Finally, because one can use any deep model to compute $f(\boldsymbol{x})$, one is free to choose one with preferred characteristics, including interpretability. For the text classification experiments, I chose a base model originally proposed for interpretable classification of medical texts (Mullenbach et al., 2018). To further unpack the explanation given for

	Income	Cover.	Lend.	Fash.	CIFAR	Subj.	Stack	IMDB	Amazon
k=1	0.85	0.76	0.96	0.95	0.96	0.98	0.81	0.95	0.78
k=5	0.91	0.77	0.98	0.97	0.98	0.99	0.84	0.98	0.87
k=10	0.93	0.77	0.99	0.98	0.98	0.99	0.85	0.98	0.90
k=100	0.96	0.83	0.99	0.99	0.99	1.00	0.89	0.99	0.94

Table 6.4: Impact of considering a subset of the training instances as an approximate explanation: the columns show agreement with the full model on the single most-probable label when basing the prediction on only the k closest training instances.

a prediction, one could, for example, inspect the attention weights for a particular pair of sentences to understand the importance of each word in context.

6.5.3 Approximate explanations

Because the weights on training instances decrease exponentially with distance, the closest training instances will contribute the most to the prediction. In some cases, only relatively few training instances will be required to fully determine the model's predicted label (because beyond this the remaining instances will lack sufficient weight to alter which class will be most highly-weighted). In practice, most test instances tend to require a substantial proportion of the nearest training instances in order to cross this threshold. However, even considering a much smaller number of the closest training instances may still result in high agreement with the prediction based on all of the data. Table 6.4 shows the agreement with the full model if one only considers the top k neighbors to each test instance. For most datasets, this agreement is very high, even for a very small number of neighbours.

6.5.4 Confidence and credibility

The above results only considered the single top-scoring label predicted for each instance; here I extend both models to the conformal setting. To verify the theoretical

expectations of conformal methods, I show that my proposed measure of nonconformity correctly works to maintain a desired error rate in making predictions. Figure 6.4 shows the results for the Fashion MNIST dataset, where I vary ε (the desired maximum error rate) from 0 to 0.2. The top subfigure (a) shows the proportion of predictions on the test set which are *correct* (that is, which contain the true label), for the softmax model using negative probability (probs) as a measure of nonconformity, DWAC using the same measure, and DWAC using my proposed measure of nonconformity (weights) given in Equation 6.13. As can be seen, all three demonstrate correct coverage, with all lines close to but not exceeding the expected proportion $(1-\varepsilon)$ across the full range (shown as a dashed line on the top subfigure). Note that this is not the same as accuracy, as some predictions may contain multiple labels.

The second and third subfigures show the same lines for the proportion of predicted label sets that are empty (b) or contain multiple labels (c). The bottom figure shows the mean number of labels in all non-empty predicted label sets. In all cases, the dashed line represents an optimal outcome (that is, a proportion of predicted label sets equal to ε are empty, all other predictions are correct, and no predictions contain multiple labels).

As can be seen, the softmax and DWAC models give indistinguishable results when using the same measure of nonconformity. My proposed measure of nonconformity, by contrast, appears to be slightly less efficient, producing slightly more predicted label sets with multiple labels, but also slightly more empty label sets, which represent identifiable errors contributing to the proportion of incorrect predictions.

The advantage of my proposed measure, however, comes in robustness to out-of-domain data. If one trains a model on the Fashion MNIST dataset, and then ask it to make predictions on the original MNIST digits dataset (which has the same size and data format, but consists of hand-written digits rather than items of clothing), one

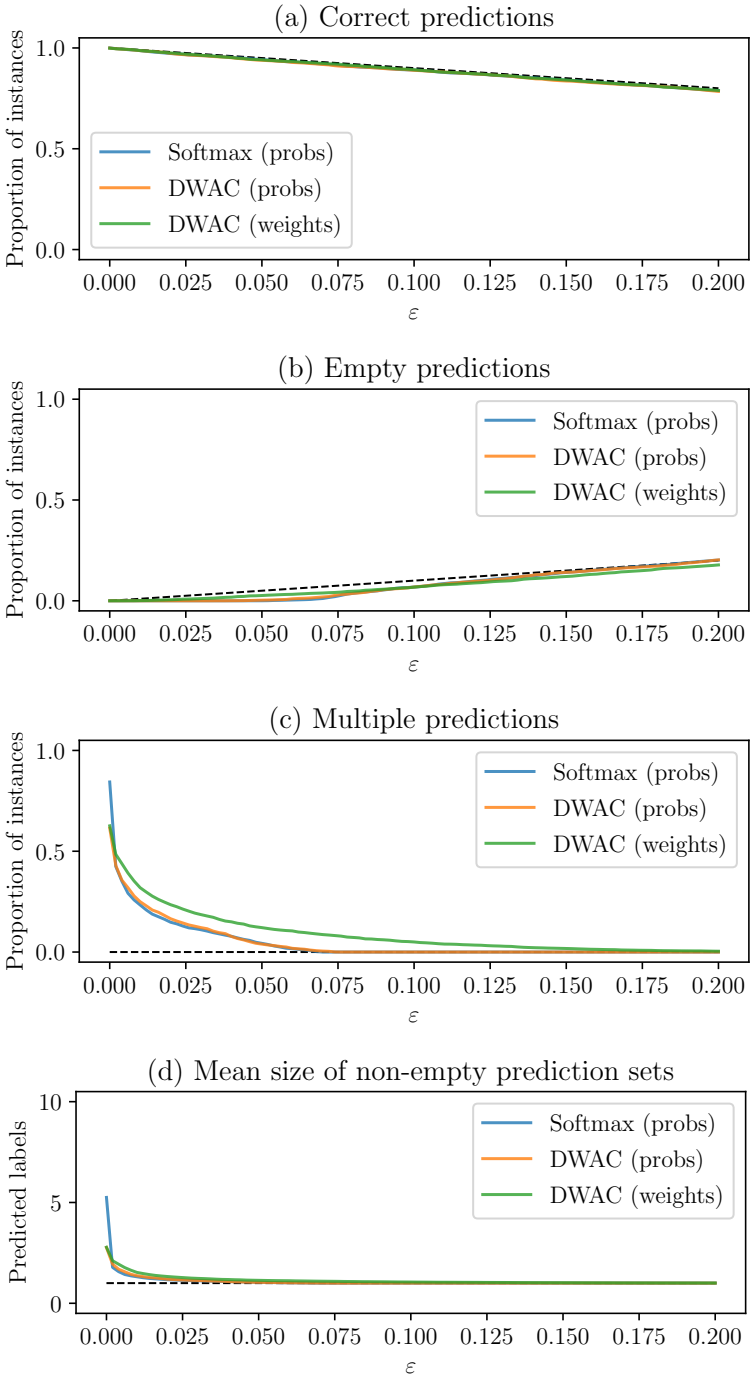


Figure 6.4: Coverage of various models on the Fashion MNIST test data, as I vary the desired maximum error rate (ε). From top to bottom, the subfigures show (a) the proportion of predicted label sets that are correct (contain the true label); (b) that are empty (make no prediction); (c) that contain multiple labels; and (d) the mean number of labels in non-empty prediction sets. The softmax and DWAC models give nearly identical results when using negative probability as a measure of nonconformity (probs). My proposed measure (weights) has an indistinguishable error rate, but is slightly less efficient. The dashed line in each figure represents an optimal response.

would hope that a good model would predict relatively low probability for all such instances. In fact, as has been previously observed (Nguyen et al., 2015a), deep models tend to predict relatively high probabilities, even for out-of-domain data, and this is also true of DWAC models.

Fortunately, the *credibility* score from a conformal predictor provides a meaningful estimate of how much one should trust the corresponding prediction.[10] Both the softmax model (using negative probability as a measure of nonconformity), and the DWAC model (using my proposed measure of nonconformity) give low credibility to the vast majority of out-of-domain examples, as shown in Figure 6.5. The credibility scores from DWAC however, are noticeably shifted closer to zero, indicating that the sum of the weights of the corresponding class is a better measure when one is concerned about the possibility of out-of-domain data. (For in-domain data, the credibility values will be approximately uniformly distributed).

I find similar results when training on CIFAR-10 images, and predicting on the Tiny Images dataset (see Figure 6.6), and an even more extreme difference in the case of the Covertype dataset, where I treat one out of seven classes as out-of-domain data, and train a model on only the six remaining classes (see Figure 6.7) . For that setup, the mean credibility score from the softmax model on the out-of-domain data is 0.9, whereas for the DWAC model with my measure of nonconformity it is 0.2. This indicates that to the softmax model, the held-out images "look like" even more extreme versions of one of the other classes, whereas the DWAC model correctly recognizes that the held-out images are relatively unlike the training instances (relative to the calibration data).

[10] Specifically, as mentioned above, it is equal to one minus the model probability that none of the labels should be given to this instance.

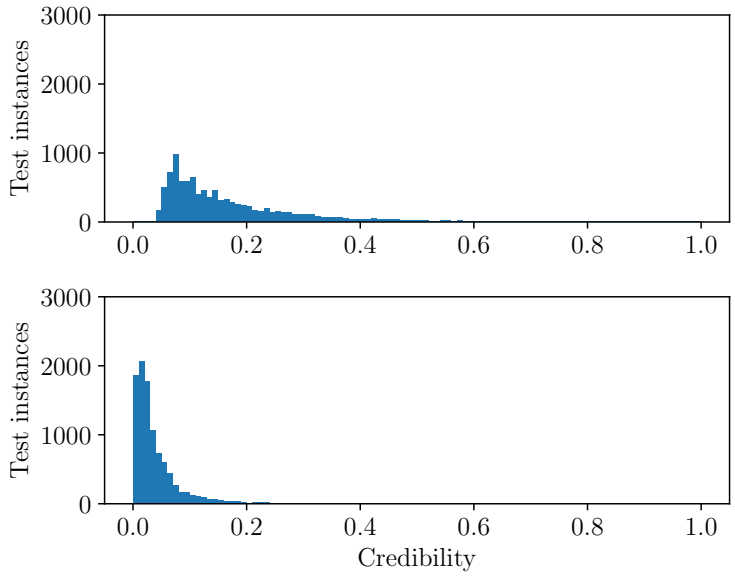


Figure 6.5: Empirical distribution of credibility scores from the softmax (top) and DWAC (bottom) models when trained on Fashion MNIST and tested on MNIST digits (which have the same input format but different content), with the latter using my proposed measure of nonconformity.

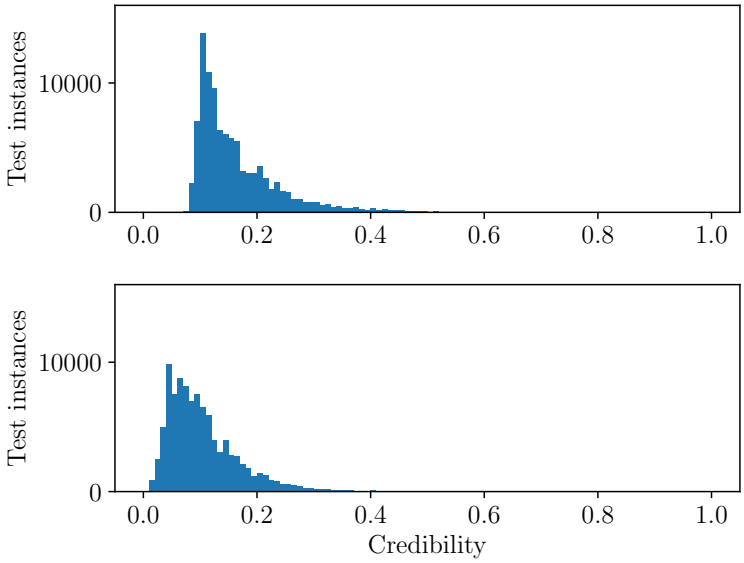


Figure 6.6: Empirical distribution of credibility scores from the softmax (top) and DWAC (bottom) models when trained on CIFAR-10 and tested on Tiny Images, with the latter using our proposed measure of nonconformity.

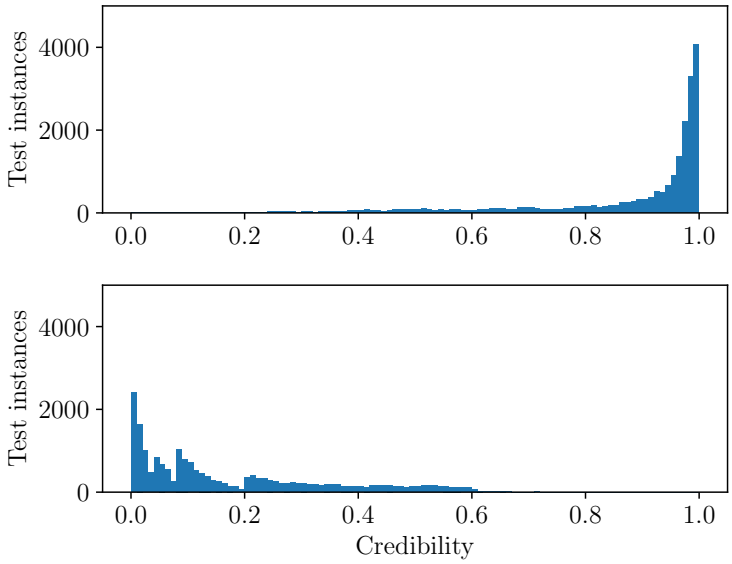


Figure 6.7: Empirical distribution of credibility scores from the softmax (top) and DWAC (bottom) models when trained on 6 of the classes in the Covertype dataset and tested on the 7th class. This represents an extreme example where the credibility scores are skewed towards 1 in the softmax model, given that in-domain data would typically be approximately uniformly distributed.

6.6 Discussion and future work

The idea of interpretability has always been important in statistics and machine learning, but has taken on a renewed urgency with the increased expressive power of deep learning models and the expanded deployment of machine learning systems in society. No single approach to interpretability is likely to solve all problems; here I have focused on adapting existing models so as to make their predictions more transparent, by decomposing them into a sum over training instances, the support for each of which can be inspected. As emphasized by papers that have made use of user studies (Huysmans et al., 2011; Kulesza et al., 2013; Narayanan et al., 2018; Poursabzi-Sangdeh et al., 2018; Yin et al., 2019), careful empirical work is required to evaluate the effectiveness of explanations, and I leave such an evaluation of this approach for future work.

A few recent papers have also sought to formalize the question of when to trust a

classifier. Jiang et al. (2018) present a method for determining how much any existing classifier should be trusted on any given test point, based on the relative distance to the most-likely and second-most-likely clusters, with clusters based on a pruned training set. This appears to be a theoretically well-motivated and empirically effective technique, but is more focused on trust than interpretability. In an unpublished paper, Papernot and McDaniel (2018) propose to train a conventional deep model, but make predictions using a k-nearest neighbours approach, with distance computed using all internal nodes of the network. My approach, by contrast, is to train a model using the same form as will be used for prediction, to make predictions based on a weighted sum over the entire training set, and rely on similarities computed in the low-dimensional space of the final layer. Wallace et al. (2018) apply the method from Papernot and McDaniel (2018) to the problem of text classification and explore the implications for interpretability.

There have also been several papers focused on the problem of predicting whether data is in-domain or out-of-domain (Lee et al., 2018; Liang et al., 2018). Many of these build on Hendrycks and Gimpel (2017), who observed that the predicted probabilities contain some useful signal as to whether data came from in-domain or out-of-domain, and proposed to use this to differentiate between the two by thresholding these probabilities. The authors did not, however, make the connection to conformal methods, which offer a more theoretically sound basis on which to make these decisions, as well as greater flexibility of metrics, beyond just predicted probability.

There are several natural extensions to this work which could be pursued, such as applying a similar architecture to regression or multi-label problems, as well as extending the idea of nonconformity to provide class-conditional guarantees.

6.7 Summary

In this chapter I have demonstrated that even for sophisticated deep learning models, it is possible to create a nearly identical model, with all of the same desirable properties, that nevertheless provides an explanation for any prediction in terms of a weighted sum of training instances. In domains where the training data can be freely inspected, this provides greater transparency by revealing the many components that explicitly contribute to a model's prediction, each of which can in principle be inspected and interrogated. Moreover, this method can build on top of other approaches to interpretability, by choosing an appropriate base model. When an approximate explanation will suffice, then using only a small subset of the training instances provides a natural high-fidelity approximation.

More importantly, representing the prediction in this manner suggests a natural alternative measure of nonconformity, which, as I have shown, provides a more effective measure for detecting out-of-domain examples. Even in cases where training data cannot be shared (due to privacy concerns, for example), this use of conformal methods still allows us to assert a quantitative estimate of the credibility of an individual prediction, one that is far more meaningful than the model's predicted probability.

The experiments in this chapter focused on a variety of data types, rather than a particular social science problem, in part to show the generality of this method. However, the ability to better interrogate why a model is making a particular prediction, or to be able to easily visualize what a model has learned, is of obvious value for model development where transparency and interpretability are important. Moreover, by providing a quantitative estimate of how much a particular prediction should be trusted, we can more effectively know the limits of where our measurements might break down, and when we should be skeptical of a model's predictions. Particularly for

social science applications, where there is real potential for the translation of ideas into policy, the credibility scores from DWACs, or from conformal methods more generally, provide a potentially useful check on the possibility of invalid predictions.

Chapter 7

Conclusion

7.1 Summary of contributions

Machine learning and natural language processing have repeatedly demonstrated their value in interdisciplinary research, and the social sciences are no exception. However, productive interdisciplinary work requires consideration of the concerns and priorities of the social sciences, including an emphasis on validity, reliability, reproducibility, interpretability, and cost. This thesis has brought together a line of work on methods for both supervised and unsupervised settings which try to take these desiderata seriously. While no single method will ever be the right solution for all problems, these chapters include examples of models designed for specific purposes, such as estimating label proportions, as well as more generically useful approaches, such as modeling documents with metadata. In all cases, the goal has been to work towards collaborative research, by incorporating thinking from the social sciences about research methods, while aiming to make machine learning and natural language processing useful to scholars from a broader range of disciplines.

Specific contributions of this thesis include:

- Chapter 3 presented a graphical model designed for automatically inferring archetypal character representations and story types in an unsupervised manner. These inferred values were shown to be useful in predicting annotations of the framing and tone of news articles about U.S. immigration, including when using a novel type of feature evaluation based on Bayesian optimization.
- Chapter 4 introduced a model which generalizes multiple topic model variants in a unified framework, using neural variational inference. In addition to showing that this allows for more flexible exploration, by allowing metadata associated with documents to be incorporated in multiple ways, this chapter also argued for using neural variational inference as the foundation of a more accessible form of modeling, with the potential to expand the range of tools available to scholars in other disciplines.
- Chapter 5 focused on the use of text classification as tool for measurement when the goal is to estimate label proportions, as is often the case in text-as-data research. It characterized two different sets of assumptions that might be made when researchers need to confront domain shift (*intrinsic* vs *extrinsic* labels), and argued that most past work targeted at accounting for this shift is based on an assumption which is inappropriate for most annotation scenarios. For the *extrinsic* setting, it was shown that the goal should be a well-calibrated model, and that we can improve upon a naive approach by using a different criterion for model selection.
- Finally, Chapter 6 demonstrated how a simple change to any deep learning classifier can create a model which is equally accurate, but with more transparently interpretable predictions, by making predictions explicitly in terms of training instances. Furthermore, this approach suggested a novel measure of

non-conformity, one which was shown to be more useful in detecting out-of-domain data. These *deep weighted averaging classifiers* (DWACs) provide a useful approach when researchers are concerned with the possibility of encountering out-of-domain instances, or want to be able to provide an *explanation* for the decisions made by a classifier.

7.2 Recurring themes

Several themes which recur throughout this work, as summarized below:

Text-as-data: As emphasized in Chapter 2, human-generated natural language holds enormous potential for learning about society. Indeed, a great deal of social science is based on text in some form, including interviews and open-ended survey responses. Historically, making use of such data required interpretation by humans to extract important insights and patterns, which is neither reproducible nor scalable. By drawing on methods from machine learning and natural language processing, we can more effectively leverage the text that is available from these sources, as well as larger-scale cultural products, such as news articles, wikis, social media, and so on. Most chapters in this thesis (especially 3, 4, and 5) proposed ways of learning from text, with the idea of *framing* providing a running example.

Exploration and measurement: Social science is typically theorized in terms of concepts (formally, constructs), but these concepts must be operationalized in terms of properties that can be measured, along with appropriate instruments. Particularly when working with complicated data such as open ended text, the measurement problem becomes especially difficult. While machine learning and natural language

processing provide many possible ways of quantifying text, there is still enormous scope for designing custom models (as in Chapter 3), as well as for expanding the space of tools that are easily useable by scholars in other disciplines (as in Chapter 4). The ideas of exploration and measurement appear throughout, but Chapter 5 is specifically concerned with the problem of using text classification as an instrument, while Chapter 6 explores how we can make such instruments more transparent and trustworthy.

Learning from social science: Just as social sciences can benefit from tools and insights derived from machine learning and natural language processing, computer science can also benefit from ideas in the social sciences. Chapter 2 provided an overview of the main concerns surrounding research methods in social science, and discussed the importance of taking these concerns seriously, especially as algorithmic systems are coming to play an increasingly prominent role in society. Chapter 3 emphasized the importance of collaborating with domain experts for theoretical grounding, validation, and interpretation of results. Other chapters addressed various properties valued by social scientists, such as reliability, calibration (Chapter 5), transparency, and interpretability (Chapter 6). For additional recent work related to this thesis addressing the issues of cost and reproducibility, please refer to Gururangan et al. (2019) and Dodge et al. (2019).

7.3 Implications for computational social science

The majority of this thesis has focused on methods, without always making explicit the connection to particular, substantive research questions in social science. This is in part to emphasize generality, but also because many social science questions merit the kind of in-depth investigation that would not easily fit into this sort of thesis (and often

warrant a thesis or a book-length treatment of their own). Nevertheless, I hope that the ideas contained within will be useful for computational social science research and the text-as-data community in particular.

Chapter 3 provided an example of an approach to exploration and measurement using probabilistic graphical models, demonstrating how a model can be designed to instantiate ideas from other fields (e.g., the idea of archetypal character types in news) and produce meaningful representations. In particular, the personas model provided a means of both discovering the sorts of character representations and story types that were present in a corpus, as well as providing a set of measurements of the corpus (i.e., as features), which were found to be predictive of framing. This sort of approach has wide-spread applicability to a variety of different types of data and questions, though as emphasized, it suffers from the need for a certain level of expertise in the area of graphical models in order to design an appropriate model and inference algorithm.

As a step towards mitigating that limitation, neural variational inference was used in Chapter 4 as a means of unifying a set of related models into a common framework, thereby providing the foundation for painless model customization by users, even by those with less expertise in statistics and machine learning. The resulting model, SCHOLAR, provides an alternative to other widely-used topic model variants, such as the structural topic model (Roberts et al., 2014), one which provides greater scalability, and the ability to incorporate richer covariates or additional prior knowledge in the form of pre-trained word vectors. As above, this sort of approach to modeling is useful primarily as a way of exploring a large collection of documents (possibly for the purpose of hypothesis generation), but also provides a way of making measurements of text (Wallach, 2016). In particular, follow on work (Gururangan et al., 2019) has demonstrated that an extension of this model is highly effective as a way to learn useful document representations for semi-supervised text classification in the limited-resources setting. Further

work is required in order to make neural variational inference a broadly accessible way of exploring text corpora, including questions of interface design and how to handle more complex models. Nevertheless, there is an opportunity here to use it as the basis of a user-friendly system for text analysis in the social sciences.

The idea of using text classification as a tool for measurement was addressed more directly in Chapter 5. This remains an important paradigm for any setting in which we would like to augment manual annotation (i.e., human coding) of documents. As shown, however, there are two important take-aways. First, if the goal is *only* to estimate the label proportions in a fixed corpus of documents, then for most applications, the *simple random sampling* approach of sampling and manually labeling a modest number of documents is likely to be far better, in terms of validity, reliability, and cost, than any approach based on dictionary methods or supervised learning. However, for cases in which we care about individual document labels, or only have access to part of the corpus during annotation, some form of supervised learning will often be necessary. As emphasized, if we must confront the problem of domain shift (as in the case of generalizing to future data), then it is important to consider the data generating process (i.e., intrinsic vs. extrinsic labels), and to recognize that validation is the only guarantee against potential error.

Finally, Chapter 6 made use of a variety of types of data, but is nevertheless highly relevant. First, in developing text classifiers as tools for measurement, it is important to be able to understand and communicate how they are working. Deep learning has brought such large gains in accuracy in numerous areas of natural language processing, that it would be ill-advised to ignore it. However, given the problems associated with deep learning as discussed in Chapter 6, it may be unappealing for many social scientists. DWACs provide, at a minimum, a more direct way of inspecting the basis for each prediction, which will help to determine if the model is doing something reasonable

during model development.

In addition, social science research is noteworthy in that many findings have the potential to be translated into policy. Any tool which is developed for the purpose of measurement (e.g., of text) has the potential to be deployed as a way of making predictions. Given the problems of domain shift discussed above, and the potential harms involved, especially when making predictions about people (e.g., based on their social media posts), it would be highly valuable to know when a model might be making a prediction that should not be trusted. The credibility scores from DWACs, or from conformal methods more generally, provide just such a mechanism; especially using DWACs, this allows us to identify data which may be unlike the training data, and for which we should be skeptical of the model's prediction.

Of course, as emphasized in the introduction, this kind of observational text-as-data research is only one part the universe of computational social science. Going beyond purely observational data, some of the ideas presented here could also be applied to the results of open-ended survey responses, for example, and could be incorporated into causal analysis in experimental work (see, for example, Fong and Grimmer, 2016).

7.4 Directions for future work

Each of the main chapters of this thesis addressed a particular problem and made a specific contribution. Nevertheless, there is great scope for potential future work.

The idea of *personas* served as the foundation for the model presented in Chapter 3, and was found to be useful for predicting annotations about the framing of news articles. Indeed, one important aspect of framing is the set of entities that are present in text, and the ways in which they are depicted. Further work along these lines seems

like a particularly promising direction for future research into framing. Although effective and interpretable, the model presented here only made use of a very limited set of textual evidence for inferring entity representations. A model which made use of greater context would be a natural extension, as would one which made use of additional metadata about the shared identity of entities depicted across documents or by different sources.

Developing a more comprehensive persona model might be difficult using traditional inference techniques, but neural variational inference, as used in Chapter 4 provides a potentially useful foundation upon which to base a model. Indeed, demonstrating that neural variational inference could easily be extended to the type of hierarchical model of text used for studying personas would be highly useful, as it would suggest greater generalizability. While there is excellent work happening in the domain of hierarchical VAE models (e.g., Sønderby et al., 2016), as well as probabilistic programming and automated inference (e.g., Tran et al., 2016), these methods have not yet seen much use in the social sciences when dealing with text.

In addition, further consideration of the trade-offs inherent in this sort of approach to inference would be valuable. Experiments in Chapter 4 suggest that inference using VAEs is inferior (in terms of model fit) to traditional collapsed Gibbs sampling with hyperparameter updating, at least for standard models like LDA, and understanding this difference would be useful. Similarly, the randomness in VAE-based inference creates potential problems in terms of reproducibility, though this could perhaps be mitigated by some sort of initialization, such as the spectral initializer used in the structural topic model (Roberts et al., 2014).

Going beyond entities, there is great scope for deeper analysis of framing, especially given the recent gains obtained on a wide range of tasks using contextual embeddings (Peters et al., 2018; Devlin et al., 2019). Promising directions include metaphor, ar-

gumentation, and narrative. Metaphor is a frequently-cited component of framing, but one which has been difficult to study computationally. Similarly, arguments are perhaps the quintessential aspect of framing (i.e., "to promote a particular problem definition, causal interpretation ...", Entman, 1993). Many codes in social science codebooks take the form of more or less specific ideas or claims (e.g., "immigration is good for the economy"), which may be expressed in many different ways. Given continued advances such as contextual embeddings, recognizing such specific ideas in text seems quite promising, given sufficient data. Finally, linking together these metaphors and claims into larger narratives which change over time is perhaps the ultimate objective, and could potentially benefit from work in computational narrative analysis.

The identification of ideas in text would be useful beyond the analysis of framing, and is a promising direction of research. A limitation, however, is the lack of evaluation data, as identifying every mention of an idea in a large corpus remains challenging. Moreover, the fact that the similarity of ideas is often implicit means that a more interactive approach, one integrating active learning and proper interface design, would allow a domain expert to navigate a corpus, gradually refining the concept by example. Additional methods of visualizing the relations between ideas, such as Tan et al. (2017), would also be useful.

Both calibration and interpretability are still underdeveloped areas which need further work. It is clear that classifier accuracy can benefit massively from pretraining (even in the low-resource setting; Gururangan et al., 2019), but we don't yet have a full appreciation of what this sort of transfer learning implies for calibration, or the potential for introducing unwanted biases into measurement. Many ideas have been proposed for interpretable models, but there is a great need for more empirical studies with real users to understand how people make sense of such information (Poursabzi-Sangdeh et al., 2018; Yin et al., 2019). Similar evaluation would be useful for thinking about

the credibility scores returned by conformal methods and the value of transparent predictions more broadly.

Finally, there is a great deal of work to be done in studying the second-order effects of these technologies. As machine learning and natural language processing are increasingly being deployed in social settings, such as recommender systems, voice assistants, and social media monitoring, we should expect that these interventions will modify the environment in which they operate. Fortunately, these fields of study also provide the means to study these kinds of downstream consequences at scale. Collaborative work in computational social science is in the best position of all fields to make sense of changes that are taking place in society due to the expansion of algorithmic interventions, and this presents a huge opportunity for further development of datasets, methods, analyses, interventions, and insights.

Bibliography

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J. Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*. 51

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. 2001. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of International Conference on Database Theory*. 103

David Aldous. 1985. Exchangeability and related topics. In *École d'Été St Flour 1983*, pages 1–198. Springer-Verlag. 36

Charles E. Antoniak. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2(6). 34

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596. 92

Josh Attenberg and Foster Provost. 2011. Inactive learning?: Difficulties employing active learning in practice. *SIGKDD Explorations Newsletter*, 12(2):36–41. 96

Ramnath Balasubramanyan, William W. Cohen, Doug Pierce, and David P. Redlawsk. 2012. Modeling polarizing topics: When do different political communities respond differently to the same news? In *Proceedings of ICWSM*. 56

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annota-

tion. In *Proceedings of EMNLP*. 96

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of ACL*. 31, 33, 35, 38

David Bamman, Sejal Popat, and Sheng Shen. 2019. An annotated dataset of literary entities. In *Proceedings of NAACL*. 51

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of ACL*. 38, 51

Pablo Barbera, Amber E. Boydstun, Suzanna Linn, Jonathan Nagler, and Ryan McMahonk. 2019. Automated text classification of news articles: A practical guide. In submission. 24

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. 2017. Interpreting blackbox models via model extraction. *CoRR*, abs/1705.08504. 99

Eric Baumer, Elisha Elovic, Ying Qin, Francesca Polletta, and Geri Gay. 2015. Testing and comparing computational approaches for identifying the language of framing in political news. In *Proceedings of NAACL*. 28, 30

Frank R. Baumgartner, Suzanna L. De Boef, and Amber E. Boydstun. 2008. *The decline of the death penalty and the discovery of innocence*. Cambridge University Press. 28

Antonio Bella, Maria Jose Ramirez-Quintana, Jose Hernandez-Orallo, and Cesar Ferri. 2010. Quantification via probability estimators. In *IEEE International Conference on Data Mining*. 77, 82

Aurélien Bellet, Amaury Habrard, and Marc Sebban. 2013. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709. 100

Shai Ben-David, Shai Shalev-Shwartz, and Ruth Urner. 2012. Domain adaptation – can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence*, 70:185–202. 80

Robert D. Benford and David A. Snow. 2000. Framing processes and social movements: An overview and assessment. *Annual Review of Sociology*, 26:611–639. 27, 28

Adrian Benton and Mark Dredze. 2018. Deep Dirichlet multinomial regression. In *Proceedings of NAACL*. 74

James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. 2015. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science and Discovery*, 8(1). 46, 47

Adam J. Berinsky and Donald R. Kinder. 2006. Making sense of issues through media frames: Understanding the Kosovo crisis. *Journal of Politics*, 68(3):640–656. 28

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. 2009. Importance weighted active learning. In *Proceedings of ICML*. 96

Anol Bhattacherjee. 2012. *Social Science Research: Principles, Methods, and Practices*. Global Text Project. 20, 21

Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2009. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10. 85

Jock A. Blackard and Denis J. Dean. 1999. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24. 113

David M. Blei. 2014. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232. 16, 19

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. 2010. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *JACM*, 57(2). 57

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of ICML*. 52, 57

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022. 19, 33, 47, 52

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Proceedings of NeurIPS*. 26

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of COLT*. 99

Jordan Boyd-Graber, Yuening Hu, and David Mimno. 2017. Applications of topic models. *Foundations and Trends in Information Retrieval*, 11(2-3):143–296. 19, 52

Amber E. Boydstun, Dallas Card, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2014. Tracking the development of media frames within and across policy issues. *Proceedings of APSA*. 29

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*. Wadsworth. 98

Jochen Bröcker. 2009. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519. 83

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computional Linguistics*, 18(4):467–479. 47

Yuri Burda, Roger B. Grosse, and Ruslan R. Salakhutdinov. 2016. Importance weighted autoencoders. *CoRR*, abs/1509.00519. 62

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186. 26

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A novel neural topic

model and its supervised extension. In *Proceedings of AAAI*. 74

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. The media frames corpus: Annotations of frames across issues. In *Proceedings of ACL*. 28

Dallas Card, Justin H. Gross, Amber E. Boydstun, , and Noah A. Smith. 2016. Analyzing framing through the casts of characters in the news. In *Proceedings of EMNLP*. 30

Dallas Card and Noah A. Smith. 2018. The importance of calibration for estimating proportions from annotations. In *Proceedings of NAACL*. 76

Dallas Card, Chenhao Tan, and Noah A. Smith. 2018. Neural models for documents with metadata. In *Proceedings of ACL*. 52, 63

Dallas Card, Michael Zhang, and Noah A. Smith. 2019. Deep weighted averaging classifiers. In *Proceedings of ACM FAT**. 97

Ali Taylan Cemgil. 2009. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, pages 4:1–4:17. 55

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of ACL*. 51

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of NeurIPS*. 55, 66

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of AAAI*. 67

Eunsol Choi, Chenhao Tan, Lillian Lee, Cristian Danescu-Niculescu-Mizil, and Jennifer Spindel. 2012. Hedge detection as a lens on framing in the GMO debates: A position paper. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*, pages 70–79. 30

Dennis Chong and James N. Druckman. 2007. Framing theory. *Annual Review of Political Science*, 10(1):103–126. 27, 28

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of CVPR*. 100

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of KDD*. 27

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297. 99

Thomas M. Cover and Peter E. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27. 99

Paul D'Angelo and Jim A. Kuypers. 2010. *Doing News Framing Analysis*. Routledge. 27

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Proceedings of NAACL*. 47

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *Proceedings of ACL*. 65

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of ICWSM*. 77

Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of ICML*. 100

Morris H. DeGroot and Stephen E. Fienberg. 1983. The comparison and evaluation of forecasters. *The Statistician: Journal of the Institute of Statisticians*, 32:12–22. 83

Dorottya Demszky, Nikhil Garg, Rob Voigt, James Zou, Jesse Shapiro, Matthew Gentzkow, and Daniel Jurafsky. 2019. Analyzing polarization in social media: Method and application to tweets on 21 mass shootings. In *Proceedings of NAACL*. 28

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*. 19, 136

Ran Ding, Ramesh Nallapati, and Bing Xiang. 2018. Coherence-aware neural topic modeling. In *Proceedings of EMNLP*. 68

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of EMNLP*. 49, 132

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *CoRR*. 22, 98

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability analysis for natural language processing: Testing significance with multiple datasets. *TACL*, 5:471–486. 22

Ellen A. Drost. 2011. Validity and reliability in social science research. *Education Research and Perspectives*, 38(1). 20

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *Proceedings of ICML*. 52, 53, 56, 60

Robert M. Entman. 1993. Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, 43(4):51–58. 13, 27, 137

Robert M. Entman. 2007. Framing bias: Media in the distribution of power. *Journal of Communication*, 57(1):163–173. 30

Michael D. Escobar and Mike West. 1994. Bayesian density estimation and inference using mixtures. *J. Amer. Statist. Assoc.*, 90:577–588. 34

Andrea Esuli and Fabrizio Sebastiani. 2015. Optimizing text quantifiers for multivariate loss functions. *ACM Trans. Knowl. Discov. Data*, 9(4). 77, 95

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89. 30

Anjalie Field, Doron Kliger, Shuly Wintner, Jennifer Pan, Daniel Jurafsky, and Yulia Tsvetkov. 2018. Framing and agenda-setting in russian news: a computational analysis of intricate political strategies. In *Proceedings of EMNLP*. 28

Casey Fiesler and Nicholas Proferes. 2018. "participant" perceptions of Twitter research ethics. *Social Media + Society*, 4(1). 26

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of EMNLP*. 51

Christian Fong and Justin Grimmer. 2016. Discovery of treatments from text corpora. In *Proceedings of ACL*. 81, 135

George Forman. 2005. Counting positives accurately despite inaccurate classification. In *Proceedings of ECML*. 77, 82, 86

George Forman. 2008. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206. 77

William A. Gameson and Andre Modigliani. 1989. Media discourse and public opinion on nuclear power: A constructionist approach. *American Journal of Sociology*, 95(1):1–37. 28

Matthew Gentzkow and Jesse M. Shapiro. 2010. What drives media slant? Evidence from U.S. daily newspapers. *Econometrica*, 78(1):35–71. 31

Todd Gitlin. 1980. *The Whole World is Watching*. Berkeley: University of California Press. 13, 27

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. Technical report. 89

Jacob Goldberger, Geoffrey E. Hinton, Sam T. Roweis, and Ruslan R. Salakhutdinov.

2004. Neighbourhood components analysis. In *Proceedings of NeurIPS*. 100

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of ICLR*. 98

Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. 2016. What does research reproducibility mean? *Science Translational Medicine*, 8(341). 22

David Graff and C Cieri. 2003. English gigaword corpus. *Linguistic Data Consortium*. 66

Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of ACL*. 30

Glenn Greenwald. 2014. *No Place to Hide: Edward Snowden, The NSA, and the U.S. surveillance state*. Picador. 31

Justin Grimmer, Solomon Messing, and Sean J. Westwood. 2012. How words and money cultivate a personal vote: The effect of legislator credit claiming on constituent credit allocation. *American Political Science Review*, 106(4). 77

Justin Grimmer and Brandon M. Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3):267–297. 12, 15, 17, 18, 24, 25, 76

Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A survey of methods for explaining black box models. *CoRR*, abs/1802.01933. 98

Odd Erik Gundersen and Sigbjørn Kjensmo. 2018. State of the art: Reproducibility in artificial intelligence. In *Proceedings of AAAI*. 22

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of ICML*. 97

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019. Variational pretraining for semi-supervised text classification. In *Proceedings of ACL*. 19, 23, 68,

74, 132, 133, 137

Eric Hardisty, Jordan L. Boyd-Graber, and Philip Resnik. 2010. Modeling perspective using adaptor grammars. In *Proceedings of EMNLP*. 30

Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *Proceedings of NeurIPS*. 20, 27

Mareike Hartmann, Tallulah Jansen, Isabelle Augenstein, and Anders Søgaard. 2019. Issue framing in online discussion fora. In *Proceedings of NAACL*. 28

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of IJCNLP*. 31

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of EMNLP*. 51

Junxian He, Zhiting Hu, Taylor Berg-Kirkpatrick, Ying Huang, and Eric P. Xing. 2017. Efficient correlated topic modeling with topic embedding. In *Proceedings of KDD*. 74

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of ICLR*. 98, 126

Philipp Hennig, David Stern, Ralf Herbrich, and Thore Graepel. 2012. Kernel topic models. In *Proceedings of AISTATS*. 59

Edward S. Herman and Noam Chomsky. 1988. *Manufacturing consent. The political economy of the mass media*. Vintage. 28, 31

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2009. Replicated softmax: An undirected topic model. In *Proceedings of NeurIPS*. 74

Daniel Hopkins and Gary King. 2010. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):220–247. 12, 15, 76, 77, 82, 83, 93

Daniel J. Hopkins and Jonathan Mummolo. 2017. Assessing the breadth of framing

effects. *Quarterly Journal of Political Science*, 12(1):33–57. 28

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H. Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of NAACL*. 95

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. 2006. Correcting sample selection bias by unlabeled data. In *Proceedings of NeurIPS*. 85

Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. 2011. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decis. Support Syst.*, 51(1). 125

Ozan İrsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of EMNLP*. 51

Mohit Iyyer, Peter Enns, Jordan L. Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of ACL*. 30

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of NAACL*. 51

Heinrich Jiang, Been Kim, and Maya R. Gupta. 2018. To trust or not to trust a classifier. *CoRR*, abs/1805.11783. 126

Michael I. Jordan and Tom M. Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. *Science*, 349:255–260. 11

John S. Justeson and Slava M. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*. 47

Daniel Kahneman. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux. 13

Katherine A. Keith and Brendan O'Connor. 2018. Uncertainty-aware generative models

for inferring document class prevalence. In *Proceedings of EMNLP*. 95

Been Kim, Cynthia Rudin, and Julie A Shah. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Proceedings of NeurIPS*. 100

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*. 114

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of ICLR*. 53, 57, 61, 62, 63

Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2017a. Human decisions and machine predictions. Technical Report 23180, National Bureau of Economic Research. 26

Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. 2015. Prediction policy problems. *American Economic Review*, 105(5):491–495. 15

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017b. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of ITCS*. 20

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. 2019. Learning hierarchical priors in VAEs. *ArXiv*, abs/1905.04982. 75

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *Proceedings of ICML*. 100

Ron Kohavi. 1996. Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In *Proceedings of KDD*. 113

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability. Technical report, University of Pennsylvania. 24

Klaus Krippendorff. 2012. *Content analysis: an introduction to its methodology*. SAGE. 18, 24, 76

Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto. 113

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. 2015. Automatic variational inference in Stan. In *Proceedings of NeurIPS*. 75

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. 2016. Automatic differentiation variational inference. ArXiv:1603.00788. 75

Todd Kulesza, Simone Stumpf, Margaret M. Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just right? ways explanations impact end users' mental models. *2013 IEEE Symposium on Visual Languages and Human Centric Computing*. 125

Brian Kulis. 2013. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364. 100

John D. Lafferty and David M. Blei. 2006. Correlated topic models. In *Proceedings of NeurIPS*. 58

Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of KDD*. 98

George Lakoff, Howard Dean, and Don Hazen. 2008. *Don't Think of an Elephant!: Know Your Values and Frame the Debate*. Chelsea Green Publishing. 27

Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. In *Proceedings of NeurIPS*. 74

Jey Han Lau, Timothy Baldwin, and Trevor Cohn. 2017. Topically driven neural language model. In *Proceedings of ACL*. 74

David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-Lszl Barabsi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, Tony Jebara, Gary King, Michael Macy, Deb Roy, and Marshall Van Alstyne. 2009. Computational

social science. *Science*, 323(5915):721–723. 11, 16

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *Proceedings of ICLR*. 126

Jing Lei, James M. Robins, and Larry A. Wasserman. 2014. Distribution free prediction sets. *J Am Stat Assoc*, 108(510):278–287. 104

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of EMNLP*. 99

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of KDD*. 31

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proceedings of ICLR*. 126

Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. 2006. Which side are you on? Identifying perspectives at the document and sentence levels. In *Proceedings of CoNNL*. 30

Zachary C. Lipton. 2016. The mythos of model interpretability. In *ICML Workshop on Human Interpretability in Machine Learning*. 22

Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible models for classification and regression. In *Proceedings KDD*. 98

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of NeurIPS*. 99

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*. 66, 113

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR*,

9:2579–2605. 116

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL*. 38, 47

Winter Mason, Jennifer Wortman Vaughan, and Hanna Wallach. 2014. Computational social science and social computing. *Machine Learning*, 95(3). 16

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of SIGIR*. 88, 113

Jon D. McAuliffe and David M. Blei. 2008. Supervised topic models. In *Proceedings of NeurIPS*. 52, 53, 55, 63

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of ICML*. 53, 55, 57, 61, 62

David Mimno and Andrew McCallum. 2008. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Proceedings of UAI*. 56, 74

Frederick Mosteller and David L. Wallace. 1964. *Inference and Disputed Authorship*. Addison-Wesley publishing company, Inc. 80

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of AAAI*. 100

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of NAACL*. 114, 119

Èlizbar A. Nadaraya. 1964. On estimating regression. *Theory of Probability and its Applications*, 9(1):141–142. 103

Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale

Doshi-Velez. 2018. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *CoRR*, abs/1802.00682. 125

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Proceedings of ACL*. 66

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 2015a. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Proceedings of CVPR*, pages 427–436. 98, 123

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015b. Improving topic models with latent feature word representations. In *Proceedings of ACL*. 65

Dong Nguyen, A. Seza Dogruöz, Carolyn Penstein Rosé, and Franciska de Jong. 2016. Computational sociolinguistics: A survey. *Computational Linguistics*, 42:537–593. 20

Khanh Nguyen and Brendan O'Connor. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of EMNLP*. 84, 112

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. 2013. Lexical and hierarchical topic regression. In *Proceedings of NeurIPS*. 31, 57

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, and Kristina Miler. 2015c. Tea party in the house: A hierarchical ideal point topic model and its application to Republican legislators in the 112th congress. In *Proceedings of ACL*. 28, 31, 52, 57

Vlad Niculae, Caroline Suen, Justine Zhang, Cristian Danescu-Niculescu-Mizil, , and Jure Leskovec. 2015. QUOTUS: The structure of political media coverage as revealed by quoting patterns. In *Proceedings of WWW*. 31

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of ICML*. 84

Helen Nissenbaum. 2009. *Privacy in Context: Technology, Policy, and the Integrity of*

Social Life. Stanford University Press. 26

Brendan O'Connor. 2014. *Statistical Text Analysis for Social Science*. Ph.D. thesis, Carnegie Mellon University. 18

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of ICWSM*. 51

Brendan O'Connor, David Bamman, and Noah A. Smith. 2011. Computational text analysis for social science: Model assumptions and complexity. In *NeurIPS Workshop on Computational Social Science and the Wisdom of Crowds*. 12, 15, 16, 18, 52, 76

Brendan O'Connor, Brandon M. Stewart, and Noah A. Smith. 2013. Learning to extract international relations from political context. In *Proceedings of ACL*. 51

Paul Ohm. 2010. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review*, 57. 26

John William Paisley, David M. Blei, and Michael I. Jordan. 2014. Bayesian nonnegative matrix factorization with stochastic variational inference. In *Handbook of Mixed Membership Models and Their Applications*, pages 205–224. 55

Sinno J. Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359. 78

Zhongdang Pan and Gerald M. Kosicki. 1993. Framing analysis: An approach to news discourse. *Political communication*, 10(1):55–75. 30

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*. 113

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135. 30

Nicolas Papernot and Patrick McDaniel. 2018. Deep k-nearest neighbors: Towards

confident, interpretable and robust deep learning. *CoRR*. 126

Martin Pelikan. 2005. Bayesian optimization algorithm. In *Hierarchical Bayesian optimization algorithm*, pages 31–48. Springer. 46

Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. 2014. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15:2009–2053. 82

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*. 19, 136

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. 85

Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna M. Wallach. 2018. Manipulating and measuring model interpretability. *CoRR*, abs/1802.07810. 125, 137

Daniel Preotiuc-Pietro, Ye Liu, Daniel Hopkins, and Lyle H. Ungar. 2017. Beyond binary labels: Political ideology prediction of twitter users. In *Proceedings of ACL*. 30

Kevin M. Quinn Burt L. Monroe Michael Colaresi Michael H. Crespin Dragomir R. Radev. 2009. How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, 54(1). 22

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2018. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444. 19

Piyush Rai, Avishek Saha, III Hal Daumé, and Suresh Venkatasubramanian. 2010. Domain adaptation meets active learning. In *Proceedings of NAACL*. 96

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009.

Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of EMNLP*. 74

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation frames: A data-driven investigation. In *Proceedings of ACL*. 51

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNNL*. 51

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna K. Jerebko, Charles Florin, Gerardo Hermosillo, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of ICML*. 95

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2018. Do CIFAR-10 classifiers generalize to CIFAR-10? *CoRR*, abs/1806.00451. 98

Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of ICML*. 53, 57, 61

Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of NAACL*. 99

Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2018a. Anchors: High-precision model-agnostic explanations. In *Proceedings of AAAI*. 99

Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2018b. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings ACL*. 99

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from Twitter. In *Proceedings of KDD*. 51

Molly Roberts, Brandon Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Gadarian, Bethany Albertson, and David Rand. 2014. Structural topic models for open ended survey responses. *American Journal of Political Science*, 58:1064–1082.

53, 133, 136

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of UAI*. 52

Francisco J. R. Ruiz, Michalis K. Titsias, and David M. Blei. 2016. The generalized reparameterization gradient. In *Proceedings of NeurIPS*. 75

Matthew J. Salganik. 2017. *Bit by Bit: Social Research in the Digital Age*, open review edition edition. Princeton University Press, Princeton, NJ. 12, 16, 23, 26, 27

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of ACL*. 26, 77

Craig Saunders, Alexander Gammerman, and Vladimir Vovk. 1999. Transduction with confidence and credibility. In *Proceedings of IJCAI*. 104, 106

Anne Schneider and Helen Ingram. 1993. Social construction of target populations: Implications for politics and policy. *The American Political Science Review*, 87(2):334–347. 31

Nathan Schneider and Noah A. Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In *Proceedings of ACL*. 38, 47

Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and abstraction in sociotechnical systems. In *Proceedings of FAT**. 27

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-CAM: Why did you say that? *CoRR*, abs/1611.07450. 99

Burr Settles. 2012. *Active Learning*. Morgan & Claypool. 96

Glenn Shafer and Vladimir Vovk. 2008. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421. 104, 106

David A. Smith, Ryan Cordell, and Elizabeth Maddock Dillon. 2013. Infectious texts: modeling text reuse in nineteenth-century newspapers. In *IEEE International Conference on Big Data*. 31

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Proceedings of NeurIPS*. 46

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP*. 95

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*. 51

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. 51

Casper K. Sønderby, Tapani Raiko, Lars Maaløe, Søren K. Sønderby, and Ole Winther. 2016. Ladder variational autoencoders. In *Proceedings of NeurIPS*. 75, 136

Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of ACL*. 51

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *Proceedings of ICLR*. 53, 55, 57, 58, 60, 62, 65, 68

Amos J. Storkey. 2009. When training and test sets are different: Characterising learning transfer. In Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Candela Sugiyama Schwaighofer Lawrence Lawrence, editors, *Dataset Shift in Machine Learning*, chapter 1, pages 3–28. MIT Press. 80

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy

considerations for deep learning in nlp. In *Proceedings of ACL*. 23

Masashi Sugiyama, Makoto Yamada, Paul von Bünau, Taiji Suzuki, Takafumi Kanamori, and Motoaki Kawanabe. 2011. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24(2). 85

Kai S. Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of ACL*. 51

Chenhao Tan, Dallas Card, , and Noah A. Smith. 2017. Friendships, rivalries, and trysts: Characterizing relations between ideas in text. In *Proceedings of ACL*. 137

Rob Tibshirani. 1996. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288. 98

Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. 2016. Edward: A library for probabilistic modeling, inference, and criticism. ArXiv:1610.09787. 75, 136

Oren Tsur, Dan Calacci, and David Lazer. 2015. Frame of mind: Using statistical models for detection of framing and agenda setting campaigns. In *Proceedings of ACL*. 28, 31

Berk Ustun and Cynthia Rudin. 2016. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391. 26, 98

Baldwin Van Gorp. 2010. Strategies to take subjectivity out of framing analysis. In Paul D'Angelo and Jim A. Kuypers, editors, *Doing News Framing Analysis: Empirical and Theoretical Perspectives*, chapter 4, pages 84–109. Routledge. 31

Vladimir Vapnik. 1998. *Statistical learning theory*. Wiley. 99

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Proceedings of NeurIPS*.

100

Vladimir Vovk. 2012. Conditional validity of inductive conformal predictors. In *Asian Conference on Machine Learning*. 106

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Springer. 104, 106

Marilyn A. Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of NAACL*. 30

Eric Wallace, Shi Feng, and Jordan Boyd-Graber. 2018. Interpreting neural networks with nearest neighbors. In *Proceedings of the Workshop on Analyzing and Interpreting Neural Networks for NLP*. 126

Hanna Wallach. 2016. Interpretability and measurement. EMNLP Workshop on Natural Language Processing and Computational Social Science. 15, 19, 22, 52, 55, 133

Hanna Wallach. 2018. Computational social science $\neq$ computer science $+$ social data. *Communications of the ACM*, 61(3). 12, 15, 16

Hanna Wallach, David M. Mimno, and Andrew McCallum. 2009. Rethinking LDA: Why priors matter. In *Proceedings of NeurIPS*. 55

Fulton Wang and Cynthia Rudin. 2015. Falling rule lists. In *Proceedings of AISTATS*. 26, 98

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of ACL*. 45

Geoffrey S. Watson. 1964. Smooth regression analysis. *Sankhya: The Indian Journal of Statistics, Series A*, 26(4):359–372. 103

Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. 2006. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of NeurIPS*. 100

Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer

learning. *Journal of Big Data*, 3(1). 78

Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. In *CoRR*. 113

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. 2002. Distance metric learning, with application to clustering with side-information. In *Proceedings of NeurIPS*. 100

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of NAACL*. 113

Yan Yan, Rómer Rosales, Glenn Fung, Subramanian Ramanathan, and Jennifer G. Dy. 2013. Learning from multiple annotators with varying expertise. *Machine Learning*, 95:291–327. 95

Ming Yin, Jennifer Wortman Vaughan, and Hanna M. Wallach. 2019. Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of CHI*. 125, 137

Dani Yogatama, Michael Heilman, Brendan O'Connor, Chris Dyer, Bryan R Routledge, and Noah A Smith. 2011. Predicting a scientific community's response to an article. In *Proceedings of EMNLP*. 64

Dani Yogatama, Lingpeng Kong, and Noah A. Smith. 2015. Bayesian optimization of text representations. In *Proceedings EMNLP*. 46

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of KDD*. 95

Jing Zhang, Xindong Wu, and Victor S. Sheng. 2016. Learning from crowdsourced labeled data: a survey. *Artif. Intell. Rev.*, 46:543–576. 95

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017.

Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the EMNLP*. 26, 83

Jun Zhu, Amr Ahmed, and Eric P. Xing. 2009. MedLDA: Maximum margin supervised topic models for regression and classification. In *Proceedings of ICML*. 74