Text Mining and Rating Prediction with Topical User Models

by

Yanir Seroussi, BSc

Thesis

Submitted by Yanir Seroussi

for fulfillment of the requirements for the degree of

Doctor of Philosophy (0190)

Supervisor: Ingrid Zukerman

Associate Supervisor: Fabian Bohnert

Clayton School of Information Technology
Monash University

August, 2012

Contents

List of Tables

List of Figures

Text Mining and Rating Prediction with Topical User Models

Yanir Seroussi, BSc
[redacted]
Monash University, 2012

Supervisor: Ingrid Zukerman
[redacted]
Associate Supervisor: Fabian Bohnert
[redacted]

Abstract

Recent years have seen an abundance of user-generated texts published online. Mining these texts for useful information is a growing research area with many aspects that are yet to be fully explored. Two such aspects, which are investigated in this thesis, are the extraction of implicit information about users to create user models, and the application of these models to tasks that require user information. Our main approach to extracting user information is via topical user models, which represent each author and document with low-dimensional distributions over topics (a topic is a distribution over words). We develop methods that utilise these topical user models to address the following tasks: (1) authorship attribution: identifying which user wrote a given anonymous text; (2) polarity inference: detecting the level of sentiment expressed in a given text; and (3) rating prediction: determining a given user's expected sentiment towards a given item.

The first task we consider is authorship attribution, where the goal is to identify the authors of anonymous texts. Authorship attribution is one of the most commonly attempted tasks in the authorship analysis field, which – in addition to authorship attribution – also deals with profiling authors by inferring demographic information and personality traits from their texts. Traditionally, research in this field has focused on formal texts, such as essays and novels, but recently more attention has been given to online user-generated texts, such as emails and blogs. Authorship attribution of online user-generated texts is a more challenging task than traditional authorship attribution, because such texts tend to be short and informal, and the number of candidate authors is often larger than in traditional settings. We address this challenge by employing topical user models. In addition to exploring novel ways of applying two popular topic models to this task, we develop a new model that

projects users and documents to two disjoint topic spaces. Employing our model in authorship attribution yields state-of-the-art performance on several datasets, which contain either formal texts or online user-generated texts, where the number of candidate authors ranges from three to about 20,000.

The second task we consider is polarity inference, where the goal is to infer the degree of positive or negative sentiment expressed in texts. Polarity inference is a key task in the sentiment analysis field, which deals with inferring people's sentiments and opinions from texts. Even though the way polarity is expressed often appears to depend on the author, most of the work in this field ignores authors. In this thesis, we introduce a framework that infers the polarity of texts by employing user-specific inference models, where the models can be weighted according to user similarity. We show that our framework outperforms two popular baselines, even when all the base models are given equal weights. In addition, we show that performance can be further improved by considering user similarity in terms of language use (e.g., as captured by topical user models) and rating patterns.

The third and final task we consider is rating prediction, where the goal is to predict the rating a given user would assign to a given item. Rating prediction is a core component of many recommender systems, which require a way to predict users' future sentiments in order to find and recommend items of personal interest. Recently, rating prediction algorithms that are based on matrix factorisation have become increasingly popular, mainly due to their high accuracy and scalability. However, such algorithms often deliver inaccurate rating predictions for users who submitted only a few ratings. In this thesis, we introduce an extension to the basic matrix factorisation algorithm that considers information about the users when generating rating predictions. We show that employing either demographic information or text-based information (in the form of topical user models) outperforms baselines that consider only ratings, thereby enabling more accurate generation of personalised rating predictions for users who have not submitted many ratings. In the case of topical user models, these predictions are generated without requiring users to explicitly supply any information about themselves and their preferences.

Text Mining and Rating Prediction with Topical User Models

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Yanir Seroussi
August 22, 2012

Acknowledgements

Following common practice, I would like to first thank my supervisors, Ingrid Zukerman and Fabian Bohnert.

Ingrid has been very welcoming and supportive ever since the first email I sent her. She convinced me to pursue a PhD rather than a masters, and helped me settle in Melbourne by stopping me from living in Clayton and inviting me to her family's Seder and Rosh Hashanah dinners (among other things). Professionally, her advice and insights have helped me complete this research project in a timely manner, as she consistently stopped me from "rearranging the deck chairs on the Titanic".

Fabian was effectively my co-supervisor from day one, even though back then he was still "only" a PhD student and my officemate. His technical advice was crucial to this project's success, and his friendship has made me feel welcome at Monash and in Melbourne in general. Even over the last year of him working from Germany, he was always there to spot holes in my arguments and supply invaluable feedback.

Monash researchers who provided help throughout this project include Russell Smyth, Mark Carman, David Albrecht and Graham Farr. Russell came up with the judgement dataset, which enabled me to get a taste of working on a real authorship attribution problem (and publish a *law* journal paper, which certainly wasn't on my radar when I started my PhD). Mark helped me get my head around all things topic modelling, even though he didn't always like the way I drew graphical models (I'm sure he would have something to say about the drawings in this thesis). David was our go-to person whenever we found ourselves faced with statistical issues that couldn't be resolved through arguments and hand-waving. Both Graham and David were on my confirmation and mid-candidature review committees, and together reassured me that I was on the right track and gave me useful feedback.

The experiments reported in this thesis could not have been run without the help of the administrators of the MSG, VPAC, MASSIVE and NCI grids. These include Philip Chan, who helped me set up my first few jobs on MSG, and was always there to answer my questions; and Paul McIntosh, who customised MASSIVE to suit my needs after I "stress-tested" it.

This research was supported by scholarships from Monash University, funds from Australian Research Council grant LP0883416, and travel grants from conferences I attended. I'm grateful for this support and to the administrative staff that helped facilitate it.

People usually thank their officemates and colleagues, and I'm not going to deviate from this custom. However, I'm worried I may forget someone, so I apologise

in advance – if I've forgotten you (and you're reading this, which is highly unlikely), it's only because you're extra-special! So, without further ado, thank you to: Adrian, Amiza, Ana Maria, Arun, Ben, Cagatay, Cora, DJ, Gideon, Greg, Hiran, Jenny, Jessie, Julie, Kerri, Marc, Marsha, Minh, Nayyar, Nomica, Patrick, Rebecca, Rokon, Steven, Su, Subrata, Thomas, Upuli, Waleed, and Yee Ling.

While not directly involved in this project, I'd like to thank my friends and family for their support. No need for lengthy lists here – you know who you are.

Last but not least, I would like to thank *The Pandy* for being super-awesome (and that's an understatement).

Chapter 1

Introduction

Recent years have seen a rapid growth in the amount of user-generated text published online. These texts take many forms: from private texts such as emails and instant messages to more public texts such as social media messages, blogs and product reviews. The information contained in such texts tends to go beyond what is explicitly expressed by the authors.[1] For example, choice of words and syntactic constructs is indicative of user demographics and personality (Argamon et al., 2009), and sentiment is often expressed implicitly rather than explicitly (e.g., saying "this movie is predictable" usually implies negative sentiment) (Pang and Lee, 2008).

Text mining and *user modelling* are two research fields that respectively deal with extracting useful information from textual data (Hearst, 1999), and building models of users with the purpose of personally tailoring the behaviour of computer systems to each user (Kobsa, 2001). Despite the huge interest in both text mining and user modelling in recent years, many aspects of these two fields are yet to be fully explored. Two such aspects, which form the focus of this thesis, are the extraction of *implicit* information from user-generated texts to create user models, and the application of these models to tasks that require user information. Our investigation of these aspects builds on and extends previous research in four main areas: topic modelling, authorship analysis, sentiment analysis and recommender systems.

Topic modelling aims to discover themes in large text corpora (Blei, 2012). This is done by defining a probabilistic representation of the latent structure of the corpus through latent factors called *topics*, which are commonly associated with distributions over words. For example, in the popular *Latent Dirichlet Allocation* (LDA) topic model, each document is associated with a distribution over topics, and each word in the document is generated according to its topic's distribution over words (Blei et al., 2003). The word distributions often correspond to a

[1] Throughout this thesis, we use the words "author" and "user" interchangeably, as the bulk of our work deals with cases where online user-generated texts are available. In addition, we omit the word "online" when discussing online user-generated texts.

human-interpretable notion of topics, but this is not guaranteed, as interpretability depends on the corpus used for training the model. Indeed, when we ran LDA on a dataset of movie reviews and message board posts, we found that some word distributions correspond to authorship style as reflected by authors' vocabulary, with netspeak words such as "wanna", "alot" and "haha" assigned to one topic, and words such as "compelling" and "beautifully" assigned to a different topic. This motivated us to use topic models to obtain compact representations of users based on their texts, yielding *topical user models*, which we define in Chapter 4 and use throughout this thesis. These topical user models aim to capture the interests of authors together with aspects of their authorship style, which is indicative of characteristics such as demographic information and personality traits. While representing user interests is a fairly straightforward application of topic models, employing topic models to also represent authorship style is one of the contributions of this thesis.

Authorship analysis deals with analysing texts in order to learn about their authors. This includes *authorship attribution*, where the main goal is to identify the authors of anonymous texts (Stamatatos, 2009), and *authorship profiling*, which deals with inferring author characteristics such as age, gender or personality traits (Argamon et al., 2009). Traditionally, research in this field has focused on formal texts, such as essays and novels, but recently more attention has been given to user-generated texts, such as emails and blogs. Authorship attribution and profiling are closely related, as similar techniques and feature types have been shown to be useful for tasks within both these areas (Argamon et al., 2009). However, obtaining data to test authorship attribution methods is much easier than obtaining data to test authorship profiling methods, since compiling an authorship attribution dataset only requires knowing the authors of the texts, while compiling a dataset for an authorship profiling task such as inferring personality traits requires the authors to answer personality questionnaires. Hence, we use authorship attribution as a testbed for topical representations of users, while addressing the inherent challenge in authorship attribution of user-generated texts, where the number of candidate authors is often large and the documents tend to be short and informal. We present the first (to the best of our knowledge) large-scale study on employing topic modelling techniques in authorship attribution, and show that our approach yields state-of-the-art performance on several datasets, which contain either formal texts or user-generated texts, where the number of candidate authors ranges from three to about 20,000 (Chapter 5).

Sentiment analysis (or *opinion mining*) deals with inferring people's sentiments and opinions from texts (Pang and Lee, 2008; Liu and Zhang, 2012). This area has received considerable attention in recent years due to the large amounts of user-generated texts available online and the applications that are enabled by the ability

to extract sentiments from such texts, such as gauging public opinion on certain issues or products. One of the main tasks in this field is *polarity inference*, where the goal is to infer the degree of positive or negative sentiment of texts (Pang and Lee, 2008). Even though the way polarity is expressed often appears to depend on the author, most of the work in this field ignores authors (Section 2.4). In Chapter 6, we address this gap by introducing a framework that considers users when performing polarity inference, by combining the outputs of user-specific inference models in a manner that makes it possible to consider user similarity (e.g., based on rating patterns or topical user models). We show that our framework outperforms two baselines: one that ignores authorship information, and another that considers only the model learned for the author of the text whose polarity we want to infer. Our experimental results support our hypothesis that the way sentiment is expressed is often author-dependent, and show that our approach successfully harnesses this dependency to improve polarity inference performance.

Recommender systems help users deal with information overload by finding and recommending items of personal interest (Resnick and Varian, 1997). While interest in recommender systems has been high since the 1990s, in recent years recommender systems have become more ubiquitous and are commercially used in various domains (Jannach et al., 2010; Ricci et al., 2011). *Rating prediction* is a core component of many recommender systems, which require a way to predict users' future sentiments in order to find and recommend items of personal interest (Herlocker et al., 1999). Recently, rating prediction algorithms that are based on matrix factorisation have become increasingly popular, due to their high accuracy and scalability (Koren et al., 2009). However, such algorithms often deliver inaccurate rating predictions for users who submitted only a few ratings (this is known as the *new user* problem). In Chapter 7, we introduce an extension to the basic matrix factorisation algorithm that considers information about the users when generating rating predictions. We show that employing either demographic information or topical user models outperforms baselines that consider only ratings, thereby enabling more accurate generation of personalised rating predictions for new users. In the case of topical user models, these predictions are generated without requiring users to explicitly supply any information about themselves and their preferences.

Figure 1.1 summarises the flow of ideas in the main chapters of this thesis. The top row specifies the research area in which each chapter's contribution lies, the middle row presents the chapters and their connections, and the bottom row shows the user aspect that is explored in each chapter. The connections between the chapters are as follows:

- The topical user models from Chapter 4 are used in different applications throughout this thesis: (1) obtaining author and document representations

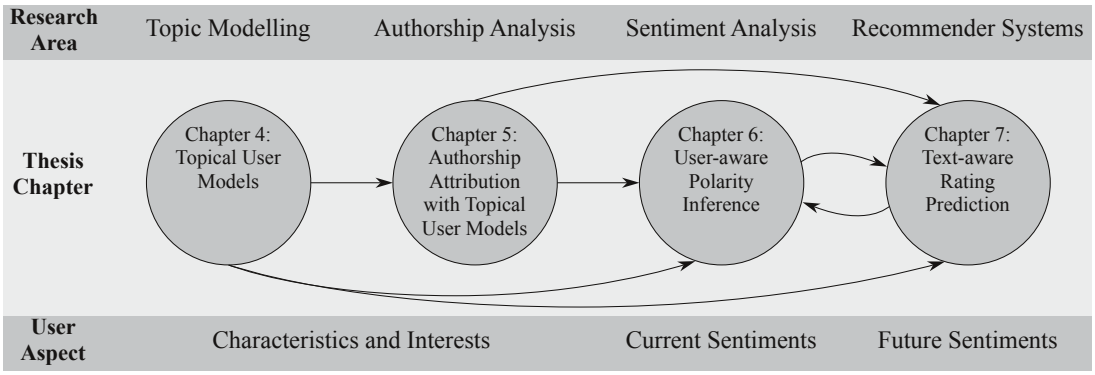


Figure 1.1: Thesis flow

and finding the most likely author of anonymous texts for authorship attribution (Chapter 5); (2) measuring user similarity for our polarity inference framework (Chapter 6); and (3) inferring text-based user attributes for our rating prediction framework (Chapter 7).

- Our polarity inference and rating prediction frameworks, which are presented in Chapters 6 and 7 respectively, rely in part on the insight that language use reflects user characteristics – an insight that originates in authorship analysis (Chapter 5).
- Chapters 6 and 7 are interconnected, since the design of our polarity inference framework is based on previous work on rating prediction, and rating prediction can be seen as a natural step forward from polarity inference – from inferring users' current sentiments to predicting their future sentiments.

In regard to the explored user aspects, topical user models are known to capture user interests (Chapter 4), and we show that they can also represent user characteristics through empirical evidence based on our authorship attribution results (Chapter 5). We then take these insights one step further through our work on polarity inference, which deals with extracting users' current sentiments (Chapter 6), and through rating prediction, where the goal is to predict users' future sentiments (Chapter 7) – in both cases these sentiments are summarised in the form of numeric ratings.

While the three tasks considered in this thesis are not new, they are approached from angles that have received little attention in the past. Specifically, to the best of our knowledge, we are among the first to publish a thorough study of the application of topic models to authorship attribution, introduce user-aware approaches to polarity inference, and consider general user texts (i.e., not only reviews) in rating prediction. This thesis builds on some of the commonalities between these tasks (such as the existence of users and user-generated texts), which come from somewhat disparate fields. We hope that our research will help inspire future work

that would further explore the connections between these fields. Exploring and utilising these connections may yield improved performance in a variety of tasks, as demonstrated throughout this thesis by our empirical results.

1.1 Thesis Outline

This thesis is structured as follows.

Chapter 2 surveys related research in the fields investigated in this thesis, focusing on topic modelling, authorship attribution, polarity inference and rating prediction.

Chapter 3 describes the experimental approach we employ throughout this thesis, together with the datasets used in our experiments.

Chapter 4 introduces the notion of *topical user models* – using topic models to obtain compact representations of users based on their texts. We discuss how two previously-suggested topic models, *Latent Dirichlet Allocation* (LDA) and the *Author-Topic* (AT) model, can be used to represent users. In addition, we present the *Disjoint Author-Document Topic* (DADT) model, which combines LDA and AT into a single model in order to separate information about the authors from information about the documents.[2] We demonstrate the merits of our DADT model in comparison to LDA and AT through experiments on a synthetic dataset.

Chapter 5 evaluates the topical user models from Chapter 4 in the context of the *authorship attribution* task.[3] We test several approaches, and show that methods based on our DADT model yield state-of-the-art performance in several scenarios where the number of candidate authors ranges from three to about 20,000. These results provide empirical evidence that topical user models retain information from user-generated texts that is representative of users' characteristics (in addition to the established finding that such models capture user interests).

Chapter 6 deals with *user-aware polarity inference*. We develop a polarity inference framework that combines the outputs of user-specific inference models. We show that even when all the models are given equal weights, our approach outperforms two baselines: one that ignores any information about the users, and another that employs only the identity of the author of the text without considering texts by other users.[4] In addition, we show that weighting the user-specific models based on user rating patterns and language use (e.g., as captured by topical user models) can improve results even further.

Chapter 7 moves from inferring user sentiments to predicting them. We introduce a *rating prediction* framework that considers user attributes when generating

[2] We first introduced DADT in (Seroussi et al., 2012).

[3] We applied LDA, AT and DADT to authorship attribution in (Seroussi et al., 2011c) and (Seroussi et al., 2012).

[4] An early version of our polarity inference framework was presented in (Seroussi et al., 2010).

rating predictions for new users with a matrix factorisation algorithm. We show that improvements in rating prediction performance are obtained by considering either demographic attributes or attributes inferred from texts using the topical user models from Chapter 4.[5]

Chapter 8 concludes the thesis and outlines possible avenues for future research.

[5]Some of our rating prediction results were reported in (Seroussi et al., 2011a).

Chapter 2

Background

This thesis is focused primarily on the extraction of implicit information from user-generated texts to create user models, and the application of these models to tasks that require user information. Therefore, this thesis draws on previous research in multiple fields, including natural language processing, data and text mining, user modelling, and information retrieval and filtering. As a comprehensive survey of these fields is beyond the scope of this thesis, we first provide a general overview of these fields, and then delve into a more detailed review of previous work on the tasks selected for this study.

This chapter is structured as follows. Section 2.1 gives a general overview of the research fields that this thesis draws on. Section 2.2 surveys topic modelling research, which serves as the basis of our topical user models (introduced in Chapter 4). Section 2.3 surveys authorship analysis and the authorship attribution task (addressed in Chapter 5). Section 2.4 presents an overview of the sentiment analysis field, focusing on polarity inference (addressed in Chapter 6). Finally, Section 2.5 reviews research on recommender systems and rating prediction (addressed in Chapter 7), and Section 2.6 summarises this chapter.

2.1 General Overview

This section briefly surveys the research fields that serve as the backdrop of the study performed in this thesis. It is worth noting that many different definitions for these fields exist, and that the lines between different research areas are often blurry. Hence, the descriptions given here are not meant to be either definitive or exhaustive. Rather, they are meant to serve as a general guide and set the scene for more in-depth discussions of the specific areas where our research lies, as described in subsequent sections.

The field of *natural language processing* groups together many tasks that pertain to the automated analysis and generation of human language (Jackson and Moulinier, 2007). Examples include parsing (Charniak, 1997), machine translation (Lopez,

2008), implementing dialogue systems (Zukerman and Litman, 2001), authorship analysis (Juola, 2006), and sentiment analysis (Pang and Lee, 2008). The rise of the web in recent decades and resultant availability of large user-generated corpora has motivated a departure from early approaches that relied on manually-crafted rules in favour of a greater focus on machine learning and statistical techniques, thereby blurring the line between natural language processing and data mining (Manning and Schütze, 1999).

Data mining aims to discover useful information and patterns from large amounts of data (Witten and Frank, 2005). Examples include analysing consumer transactions to find trends (Agrawal et al., 1993), generating forecasts from weather observations (Basak et al., 2004), and automated clustering of textual data according to topics (Blei, 2012). Like natural language processing, data mining often involves the use of statistical and machine learning techniques (Witten and Frank, 2005). Moreover, the sub-area of *text mining*, which deals with mining information from textual data (Hearst, 1999), can be seen as the intersection between natural language processing and data mining. For example, sentiment analysis is defined by some researchers as a text mining task. Indeed, the term *opinion mining* is commonly seen as synonymous to sentiment analysis (Pang and Lee, 2008).

User modelling is a research field that focuses on building models of users with the purpose of personally tailoring the behaviour of computer systems to each user (Kobsa, 2001). The user modelling field can be seen as partially reliant on natural language processing and data mining, as techniques from these two fields are often required to build user models. For example, natural language processing techniques stand at the core of dialogue systems that adapt themselves to specific users (Zukerman and Litman, 2001), and data mining methods are required to personalise the information presented to users online given overwhelming amounts of data, e.g., via recommender systems (Resnick and Varian, 1997). However, user models do not have to be tailored to a specific task, and some researchers focus on the creation of generic user models (Kobsa, 2001), with the grand vision of lifelong user modelling (Kay and Kummerfeld, 2009).

The goal of *information retrieval* systems is to retrieve relevant information from large amounts of data, often in response to an explicit user query (Salton and McGill, 1983). The related area of *information filtering* aims to expose relevant information in a way that is based on a representation of long-term user needs (Hanani et al., 2001). With the advent of the web, information retrieval systems have become ubiquitous, to the point where performing web searches is a part of everyday life for most people in the modern world (Manning et al., 2008). Information filtering systems are also commonplace, e.g., in the form of online personalised advertisements and recommendations that are meant to increase user engagement (and also generate revenue

for commercial websites) (Ricci et al., 2011). Information retrieval and filtering are closely related to the areas of natural language processing and data mining: much of the information to be retrieved or filtered is given in human language, in raw data form, or in a combination of the two, e.g., information about consumers' past transactions combined with user-generated product reviews can be used to drive recommender systems (Pang and Lee, 2008). In addition, information retrieval and filtering systems often personalise their output based on user models (Kobsa, 2001).

The research areas we survey in the remainder of this chapter fall within one or more of the above fields:[1]

- *Topic modelling* (Section 2.2) can be seen as belonging to the natural language processing and data mining fields, and is a key component of many information retrieval systems. Throughout this thesis, we show that the topic models introduced in Chapter 4 can be used for user modelling tasks (we refer to these models as *topical user models*).
- Modern *authorship analysis* (Section 2.3) has some of its roots in natural language processing (though research in this area predates modern computing), and can be seen as a collection of text mining tasks. With more people generating texts than ever before, we believe that authorship analysis will become an important part of user modelling, which enables personalised information retrieval and filtering. Hence, we use the authorship attribution task as a testbed for topical user models (Chapter 5).
- As mentioned above, *sentiment analysis* (Section 2.4) is a research area that lies at the intersection between natural language processing and data mining. This area has garnered much attention in recent years due to the abundance of sentiment-bearing texts found online. Sentiment analysis has also received attention from the information retrieval and filtering communities, e.g., as a component of systems that return documents that express a certain sentiment (Aciar et al., 2007). A gap in sentiment analysis research, which we address in Chapter 6, is the lack of attention to author identity when analysing sentiment in user-generated texts.
- Finally, *recommender systems* (Section 2.5) are a prime example of information filtering, and as such are often strongly dependent on data mining and user modelling. Most recommender systems that make use of natural language processing do so only when the subject matter of the recommendations is textual (Lops et al., 2011). Following our goal of modelling users based on their texts, our main contribution in Chapter 7 is presenting a way of harnessing natural language processing in the form of topical user models to improve the

[1] Most references are omitted here, as they are detailed in subsequent sections.

accuracy of rating predictions for new users. These predictions can potentially form the basis of personalised recommendations.

2.2 Topic Modelling

Recent years have seen a growing interest in the design and application of topic models, which aim to discover themes in large text corpora (Blei, 2012). Interest in topic models has been fuelled by their flexibility and applicability to a wide range of tasks that go beyond theme discovery. These tasks include document dimensionality reduction (Blei et al., 2003), polarity inference (Zhu and Xing, 2010), rating prediction (Shan and Banerjee, 2010), and even video analysis (Hospedales et al., 2009).

One of the areas we explore in this thesis is the notion of *topical user modelling* – using topic models to characterise users based on texts they authored. Hence, our focus in this section is on research that we found relevant to this direction. Specifically, we start by discussing the models that are widely perceived to have sparked the recent interest in topic modelling (Section 2.2.1). Then, we survey approaches to integrating metadata into topic models, focusing on user-related metadata (Section 2.2.2). We round off this section with a brief review of research on accounting for word order in topic models (Section 2.2.3), and approaches commonly used to evaluate topic models (Section 2.2.4).

2.2.1 Seminal Models

Recent interest in topic modelling is commonly seen to have stemmed from the introduction of the *Latent Dirichlet Allocation* (LDA) model (Blei et al., 2001, 2003). The main idea behind LDA is that each document in a corpus is associated with a distribution over topics, and each word in the document is generated according to its topic's distribution over words. One of the motivations behind LDA was to address the limitations of an earlier model, *probabilistic Latent Semantic Indexing* (pLSI) (Hofmann, 1999), which does not offer a natural way to infer the topics of new documents and is also prone to overfitting (Blei et al., 2003). The pLSI model is a probabilistic version of the *Latent Semantic Indexing* (LSI) (Deerwester et al., 1990) model, which employs singular value decomposition to reveal the latent structure of the document word-frequency matrix.

In this thesis, we use LDA directly to model users and also as one of the building blocks of our *Disjoint Author-Document Topic* (DADT) model (Section 4.3). Hence, we briefly discuss LDA and some of its extensions here, and provide a more in-depth technical discussion in Section 4.2.

An important point to clarify is the meaning of the word "topic". Topics in LDA and other topic models are latent factors, which are commonly associated

with distributions over words.[2] These distributions often correspond to a human-interpretable notion of topics, but this is not guaranteed, as interpretability depends on the corpus used for training the model. A common way of visualising topics is by displaying lists of words that account for most of the probability density in each topic (Blei, 2012). If, for example, stopwords are not filtered out from a corpus of English documents, the top words in each topic are likely to be function words like "the" and "and", which cannot be seen to be associated with human-interpretable topics (but can still be used as markers of authorship style, since their frequency in each topic varies). It is worth noting that even if stopwords are discarded, nothing forces the word distributions to stand for actual topics. Indeed, when we ran LDA on a corpus of movie reviews and message board posts (where stopwords were discarded), we found that some word distributions corresponded to interpretable topics, with words such as "noir" and "detective" considered to be highly probable for one topic. However, other word distributions seemed to correspond to authorship style as reflected by authors' vocabulary, with netspeak words such as "wanna", "alot" and "haha" assigned to one topic, and words such as "compelling" and "beautifully" assigned to a different topic. These observations motivated us to use topic models to obtain compact representations of users based on their texts (Chapter 4).

One challenge posed by the probabilistic structure of topic models is that accurate inference of the models is intractable, and thus efficient approximation algorithms need to be developed. The two main ways of addressing this problem are based on either variational inference (Blei et al., 2001, 2003), or Gibbs sampling (Griffiths and Steyvers, 2002a,b, 2004). We follow the Gibbs sampling approach due to its efficiency and ease of implementation (Steyvers and Griffiths, 2007), and provide a detailed account of the inference algorithms for the models we use in Chapter 4.

The modularity and flexibility of LDA yielded a plethora of extensions to the basic model. Some of the basic extensions to LDA include placing priors on the word distributions (Griffiths and Steyvers, 2002a,b, 2004) (discussed in Section 4.1.2), automatically inferring the number of topics rather than specifying it in advance (Teh et al., 2006), and optimising the priors of the model (Wallach et al., 2009a). Survey papers that discuss these issues in greater depth include (Steyvers and Griffiths, 2007; Blei and Lafferty, 2009; Blei, 2012). In the next two sections we discuss extensions to LDA that are relevant to our focus on forming topical user models for the authors of user-generated texts: integrating metadata and accounting for word order.

[2]It is worth noting that such topics are different from the topics considered in *supervised* topic identification, where the topics are predefined category labels (Bigi et al., 2001; Joachims, 1998).

2.2.2 Integrating Metadata

Since the introduction of LDA, many extensions that integrate different types of metadata have been proposed (Blei, 2012). An example that is of particular interest to us is the *Author-Topic* (AT) model (Rosen-Zvi et al., 2004), whose original aim was to model the interests of authors in corpora of multi-authored texts (e.g., research papers). In contrast to LDA, AT generates each document in a corpus from its *authors'* distributions over topics, rather than from a document-specific topic distribution. We use AT to model users and also as one of the building blocks of our DADT model. Hence, we discuss the technical details of AT in Section 4.2.

AT can be seen as belonging to a large family of topic models that incorporate metadata, named *upstream* models by Mimno and McCallum (2008). Upstream models constrain the topic distributions according to the metadata labels, i.e., the labels are placed *above* the topics in the generative process. The other family of models identified by Mimno and McCallum is that of *downstream* models, which generate the labels from the topic assignments, i.e., the labels are placed *below* the topics in the generative process.

Many models that are tailored to *task-specific* metadata have been defined. Examples of task-specific upstream models include Mimno and McCallum's (2007) model that extends AT to address cases where authors write under several personae, Hospedales et al.'s (2009) model that sees actions in videos as topics and constrains them according to long-term behaviours, and Sauper et al.'s (2011) model that discovers product properties and sentiments towards these properties by drawing review snippet words conditionally on both properties and sentiment attributes. Examples of task-specific downstream models include Erosheva et al.'s (2004) model that analyses scientific papers by generating each paper's abstract and reference list from the same set of topics, Newman et al.'s (2006) model that discovers named entities in texts by conditioning them on the topics, and Guo and Diab's (2011) model that draws word senses according to the topic distributions and then generates each word according to its chosen sense.

A line of research that has garnered much interest in recent years is the definition of *generic* models that incorporate metadata (Blei and McAuliffe, 2007; Mimno and McCallum, 2008; Ramage et al., 2009; Zhu and Xing, 2010). Such models have the appealing advantage of obviating the need to define a new model for each new task (e.g., generic models may potentially be used to generate author representations by defining a metadata label for each author). However, this advantage may come at the price of increased computational complexity or poorer performance than that of task-specific models (Mimno and McCallum, 2008). As the focus of this thesis is on modelling users based on their texts, we experimented only with LDA and with

the task-specific topic models discussed in Chapter 4 (AT and DADT, which model authors explicitly). The applicability of generic models to the tasks considered in this thesis is an open question that would be interesting to investigate in the future. Nonetheless, most of the generic models surveyed here have limitations that make them unsuitable for our purposes.

Examples of generic upstream models include *Discriminative LDA* (DiscLDA) (Lacoste-Julien et al., 2008), *Labeled LDA* (Ramage et al., 2009) and *Dirichlet-Multinomial Regression* (DMR) (Mimno and McCallum, 2008). DiscLDA and Labeled LDA dedicate at least one topic to each metadata label, making them too computationally expensive to use on datasets with thousands of authors, such as the Blog and IMDb1M datasets (described in Section 3.3). In contrast to DiscLDA and Labeled LDA, DMR uses less topics by sharing them between labels. Mimno and McCallum (2008) showed that DMR outperformed AT on authorship attribution of multi-authored documents (in a dataset of research papers where stopwords were filtered out). Despite this, we decided to use AT, since we found in preliminary experiments that AT performs better than DMR on authorship attribution of single-authored texts – such texts are the main focus of this thesis.[3] It is worth noting that we retained stopwords in our experiments, because they are known to be indicators of authorship (Section 2.3.2).

A representative example of a generic downstream model is *supervised LDA* (sLDA) (Blei and McAuliffe, 2007), which generates labels from each document's topic assignments via a generalised linear model. This model was extended by Zhu et al. (2009), who introduced *Maximum entropy discrimination LDA* (MedLDA), where model training is done in a way that maximises the margin between labels, which is "arguably more suitable" for inferring the labels of unseen documents. Zhu and Xing (2010) further extended that work by introducing *supervised Conditional Topic Random Fields* (sCTRF), which combines sLDA with conditional random fields to accommodate arbitrary types of features. Zhu and Xing applied these models to the polarity inference task, and found that support vector regression outperformed sLDA and performed comparably to MedLDA (these three models employed only unigrams), while sCTRF yielded the best performance by incorporating additional feature types (e.g., part-of-speech tags and a lexicon of positive and negative words). Based on these results, we decided to leave experiments with downstream models for future work, as it seems unlikely that we would obtain good results on the authorship attribution task without considering other feature types in addition to token unigrams (which is beyond the scope of this thesis).

[3]We briefly consider multi-authored *training* texts in the authorship attribution experiments presented in Section 5.4, but in that case the anonymous *test* texts are single-authored. In all our other experiments, we consider only datasets of single-authored texts (Section 3.3).

2.2.3 Accounting for Word Order

Many topic models work under the *word exchangeability* (or bag of words) assumption (Blei, 2012). That is, they ignore the order of the words in the given documents. While this assumption simplifies model inference, it inevitably causes the loss of some information. For example, LDA may find topics where the words "united" and "states" are very likely to occur, but it cannot discover the more meaningful combination of "united states". From our perspective, word order may be of importance as an indicator of author style, since, e.g., authors from different demographic groups may use different syntactic constructs and idioms.

Several models that address the issue of word order have been suggested. An early example is Griffiths et al.'s (2004) model, which combines LDA with a *hidden Markov model* (HMM) (Rabiner and Juang, 1986) by drawing a class for each word (based on the class transition probabilities from the previous word's class), and either generating the word as in LDA (if the class number is equal to 1), or drawing it directly from the word distribution of the word's class (if the class number is not equal to 1). Other models include those by Wallach (2006), Wang et al. (2007), Gruber et al. (2007), Zhu and Xing (2010) and Sauper et al. (2011). Wallach (2006) extended LDA by making each word dependent on both its topic and on the previous word. This was done by defining a large number of distributions over words (LDA has only T word distributions, while Wallach's model has $T \times V$ word distributions, where T is the number of topics and V is the size of the vocabulary). Wang et al. (2007) extended Wallach's (2006) work by allowing some words to be generated as unigrams while others could be generated as parts of n-grams in a manner that depends on both the topics and the previous words (Wang et al.'s model has more parameters than Wallach's model). Like Griffiths et al. (2004), Gruber et al. (2007) combined LDA with HMMs. However, Gruber et al. used HMMs to model their assumption that all the words in the same sentence are drawn from the same topic, while consecutive sentences are likely to be about similar topics. Similar approaches were taken by Zhu and Xing (2010) and Sauper et al. (2011), who used conditional random fields and HMMs respectively to model topic dependencies between consecutive words.

Models such as those introduced by Wallach (2006) and Wang et al. (2007) appear to be too computationally expensive to infer on large corpora due to their large number of parameters: each word distribution contains V elements, and the number of word distributions is $T \times V$, meaning that the number of parameters grows quadratically with the size of the vocabulary (which may comprise hundreds of thousands of words). On the other hand, the other models mentioned above (Griffiths et al., 2004; Gruber et al., 2007; Zhu and Xing, 2010; Sauper et al., 2011) seem better

suited to large amounts of data, as the number of word distributions is independent of the size of the vocabulary. This is why we chose Griffiths et al.'s (2004) model as the basis of our attempt to consider both document authors and word order, which is described in detail in Section 4.5.2.

2.2.4 Evaluation of Topic Models

The three main aspects of topic models that are commonly evaluated are: (1) probability of held-out data, (2) interpretability, and (3) performance on a specific task.

The idea behind measuring the *probability of held-out data* is to test how well the models generalise given the training data. While this approach has been used by many researchers (e.g., Blei et al., 2003; Griffiths and Steyvers, 2004; Rosen-Zvi et al., 2004), it has been criticised for not testing whether the models achieve their goal of uncovering the latent structure of the training corpus (Chang et al., 2009; Mimno and Blei, 2011; Blei, 2012). Moreover, reliably estimating the probability of held-out data is far from simple, as exemplified by Wallach et al.'s (2009b) empirical comparison of methods for estimating this probability.

Measuring model *interpretability* generally requires some subjective human analysis of the inferred topics (Chang et al., 2009). Nonetheless, many studies present some examples of the inferred topics as a means of qualitative analysis (e.g., Blei et al., 2003; Griffiths and Steyvers, 2004; Rosen-Zvi et al., 2004). Chang et al. (2009) attempted to formalise this analysis by performing large-scale human studies, and showed that interpretable models are not always the ones that exhibit the best performance in terms of probability of held-out data. Mimno and Blei (2011) suggested a way of automating this process by drawing on model checking techniques from Bayesian statistics, but did not compare their approach to results from human studies, such as those done by Chang et al. (2009).

Task-specific evaluation involves using the models for the task at hand and measuring performance in a model-independent manner (Lu et al., 2011b). For example, one of the ways Blei et al. (2003) chose to evaluate their LDA model was to use topic distributions as low-dimensional representations of documents in a corpus, and use these distributions as input to a support vector machine classifier. In this case, Blei et al. compared the classification accuracy (i.e., the percentage of documents classified correctly) obtained with LDA-based representations to the accuracy obtained with word frequency features.

Our main focus in this thesis is on the tasks of authorship attribution, polarity inference and rating prediction. Therefore, we evaluate the topic models presented in Chapter 4 in the context of these tasks (Chapters 5, 6 and 7). In addition, we offer some limited analysis of model interpretability in Section 4.4 (further interpretability analysis would require user studies, which are beyond the scope of this thesis). We

do not follow the evaluation approach of measuring the probability of held-out data, since we use topic models to represent users, and models that assign high probability to held-out data are not guaranteed to yield good user representations.

2.3 Authorship Analysis

Authorship analysis deals with analysing texts in order to learn about their authors. This includes *authorship attribution*, where the main goal is to identify the authors of anonymous texts (Stamatatos, 2009), and tasks that may be grouped under *authorship profiling*, which deal with inferring author characteristics such as age, gender or personality traits (Argamon et al., 2009). Authorship attribution and profiling are closely related, as similar techniques and feature types have been shown to be useful for tasks within both these areas (Argamon et al., 2009). Indeed, some researchers define authorship profiling as an attribution task (Juola, 2006), while others consider authorship profiling to be a natural – but distinct – extension of attribution (Argamon et al., 2009). To avoid confusion, we consistently use the term *analysis* to group the distinct areas of *attribution* and *profiling*.

Our main contribution to the authorship analysis field is in being among the first to apply topic modelling techniques to authorship attribution. We show in Chapter 5 that authorship attribution methods based on the topical user models from Chapter 4 yield state-of-the-art performance in several scenarios where the number of candidate authors ranges from three to about 20,000, with the best results in most cases obtained by methods based on our DADT model. This suggests that topical user models can capture user characteristics as reflected by user-generated texts due to the strong correlation between authorship attribution and profiling performance. Empirical evidence supporting this hypothesis is supplied by our successful application of topical user models to the polarity inference and rating prediction tasks, since the way users express polarity and the ratings they give to certain items are correlated with their characteristics (Chapters 6 and 7 respectively).

In this section, we first survey authorship analysis tasks, focusing on recent developments in the field (Section 2.3.1). We then provide a brief overview of feature types that were found to be indicative of authorship (Section 2.3.2), and end with a discussion of the methods used by researchers in the field (Section 2.3.3).

2.3.1 Task Overview

Authorship attribution has a long history that predates modern computing. For example, Mendenhall (1887) suggested in the end of the nineteenth century that word length can be used to distinguish works by different authors. Modern interest in authorship attribution is commonly traced back to Mosteller and Wallace's (1964)

study on applying Bayesian statistical analysis of function word frequencies to uncover the authors of the *Federalist Papers* (Juola, 2006; Koppel et al., 2009; Stamatatos, 2009). This sustained interest in authorship attribution is due to its many applications in various areas, such as computer forensics, criminal law, military intelligence, and humanities research. In recent years, authorship attribution research has been fuelled by advances in natural language processing, text mining, machine learning, information retrieval, and statistical analysis. This has motivated the organisation of workshops and competitions to facilitate the development and comparison of authorship attribution methods (Juola, 2004; Argamon and Juola, 2011).

The focus of our work is on extracting information from user-generated texts, with the goal of creating user models that are applicable to various tasks beyond authorship attribution. Hence, in this survey we focus mainly on general trends and recent developments in the authorship attribution area. Recent comprehensive historical surveys include those by Juola (2006), Koppel et al. (2009) and Stamatatos (2009).

Argamon and Juola (2011) grouped authorship attribution tasks into three categories in their summary of the authorship attribution competition at the PAN'11 Workshop:

- *Closed-set attribution*: Training texts by the candidate authors are supplied in advance. For each test text, the task consists of attributing the text to the correct author out of the candidate authors.
- *Open-set attribution*: Training texts by the candidate authors are supplied in advance. For each test text, the task consists of attributing the text to the correct author out of the candidate authors, or determining that it was written by none of the candidate authors.
- *Verification*: Training texts by only *one* candidate author are supplied in advance. For each test text, the task consists of verifying whether it was written by the candidate author or not.

While most of the early authorship attribution research focused on the closed-set task, recent years have seen a growing interest in open-set attribution and verification (Koppel and Schler, 2004; Sanderson and Guenter, 2006; Koppel et al., 2011b). Completely unsupervised decomposition of multi-authored texts into authorial units has also been studied (Koppel et al., 2011a). In this thesis, we consider only closed-set attribution, but our methods can be extended to account for other authorship attribution scenarios (Section 8.2).

Regardless of the underlying task, another challenge currently faced by researchers in the field is dealing with informal user-generated texts, which tend to be shorter and with more grammar and spelling errors than formal texts (Argamon and

Juola, 2011; Koppel et al., 2011b). The corpus chosen for the PAN'11 competition, which contains relatively short emails (less than 100 tokens on average) by tens of authors (Argamon and Juola, 2011), illustrates this challenge. Another example comes from Koppel et al.'s (2011b) work on a corpus of blog posts by thousands of authors. Our approach to authorship attribution addresses this challenge by employing topic models, which have been shown to be successful in dealing with large amounts of informal text (Section 2.2).

Authorship profiling is a relatively new area of research, which is directly connected to authorship attribution, because similar feature types and methods can often be used for both authorship profiling and attribution (Argamon et al., 2009). Authorship profiling tasks include identifying author gender (Koppel et al., 2003; Schler et al., 2006; Sarawgi et al., 2011), age (Koppel et al., 2003; Schler et al., 2006; Rosenthal and McKeown, 2011), native language (Koppel et al., 2005; Wong et al., 2011), and personality type (Argamon et al., 2005; Oberlander and Nowson, 2006; Iacobelli et al., 2011). One aspect of these problems that can make them somewhat easier than authorship attribution is that the classification categories are well-defined and their number is relatively small, unlike authorship attribution, which may involve hundreds or thousands of candidate authors. On the other hand, conflicting writing styles between authors that share the same characteristics (e.g., authors of the same gender) may make some profiling tasks harder than authorship attribution. While we do not consider profiling tasks directly, it is important to keep in mind that authorship attribution performance and authorship profiling performance are correlated, and it is likely that our topic modelling approach can be successfully applied to profiling tasks (Section 8.2). Moreover, the insight that language use is indicative of demographics and personality is what motivated some of our user similarity models for polarity inference (Sections 6.4.4 and 6.4.5), and our approach to deriving text-based attributes for rating prediction (Section 7.3.3).

2.3.2 Features Indicative of Authorship

Features that are commonly used in authorship analysis range from "shallow" features, such as token and character n-gram frequencies, to features that require deeper linguistic analysis, such as part-of-speech and rewrite rule frequencies (Stamatatos, 2009). In addition, some datasets lend themselves to the extraction of corpus-specific features, e.g., Tanguy et al. (2011) used the terms in the openings and closings of emails as separately-weighted features since they observed that some authors repeat these segments in all their emails.

It is often the case that good performance can be obtained by relying only on token unigrams, despite their simplicity (Koppel et al., 2009). An interesting point is that stopwords (particularly, function words and punctuation) are considered to

be indicative of authorship, which stands in contrast to other text classification tasks (e.g., categorisation by predefined topic labels). Indeed, using stopwords as features has been advocated as a way of extracting content-independent authorship markers (Gamon, 2004a). However, we note that even pronouns such as "she" can be seen as carrying content-related information. For example, a novel with primarily female characters is more likely to contain the word "she" than a novel with only male characters, which may be the reason why Koppel et al. (2003) found that the word "she" is highly indicative of female authorship when classifying fiction works by author gender. In addition, filtering out uncommon words may result in poor performance, as any author (or group of authors in the case of authorship profiling) is likely to have some "favourite" uncommon words, regardless of content.

An interesting example of a feature type that requires sophisticated analysis is that of "unstable" words, which Koppel et al. (2006) defined as words that can easily be replaced by other words. They obtained a list of unstable words by automatically translating text from English to other languages and then back to English. The words that remained the same in the translated texts were considered stable (e.g., "and" and "the"), while words that changed were assigned an instability score according to the number of texts where they changed (e.g., "over" and "above" were found to be highly unstable). The main problem with using unstable words is their language-dependence, which means that they have to be found separately for each language. In addition, using different machine translation algorithms and different corpora may result in different lists of unstable words.

Other examples of feature types can be found in recent surveys of authorship analysis research (Juola, 2006; Stamatatos, 2009; Koppel et al., 2009; Argamon et al., 2009; Argamon and Juola, 2011). A recent example of a successful integration of many feature types is described in Tanguy et al.'s (2011) aptly-titled paper (*A Multitude of Linguistically-rich Features for Authorship Attribution*), which describes their submission to the PAN'11 competition that obtained the first place in several scenarios (Argamon and Juola, 2011). The feature types employed include suffix use, consecutively occurring punctuation marks (e.g., *!!!*), emoticons, capitalisation patterns, morphological complexity, spelling errors, US/UK variations, named entities, syntactic complexity, and semantic cohesion as calculated from semantic similarity between the words in the text. Unfortunately, Tanguy et al. did not offer an analysis of the contribution of each feature type to the overall performance. Moreover, as many of their features were extracted using external data sources (e.g., WordNet and the CELEX database), their results may be hard to reproduce, especially on datasets of non-English texts. Nonetheless, Tanguy et al.'s work opens the door to follow-up studies that would provide better insights into the types of features that are useful in various authorship analysis scenarios.

2.3.3 Authorship Analysis Methods

Recent years have seen a move from traditional authorship attribution methods, which often required manual analysis of the data, to large-scale automated machine learning methods (Juola, 2006). As for other text classification tasks, *Support Vector Machines* (SVMs) have been shown to deliver good performance, because they can handle feature vectors of high dimensionality (Stamatatos, 2009; Koppel et al., 2009). Hence, SVM performance often serves as a baseline for other methods. In addition, other popular methods such as maximum entropy and logistic regression have been shown to yield competitive performance (Tanguy et al., 2011; Kern et al., 2011).

Interest in cases where the test texts may have not been written by any of the candidate authors (i.e., open-set attribution and verification, described in Section 2.3.1) has led to the development of techniques that specifically handle such cases. A fairly straightforward approach in the case of probabilistic methods is to set a threshold on the probability assigned to the selected author based on performance on held-out data – if the probability of the selected author is below the threshold, then "unknown author" is returned. This approach was successfully employed by Tanguy et al. (2011) in conjunction with a maximum entropy classifier. An alternative approach to authorship verification was introduced by Koppel and Schler (2004), who exploited the observation that eliminating highly weighted features when verifying the authorship of a given text will result in large performance degradation only for the true author. Hence, they iteratively eliminated the highest-weighted features from the training set and employed SVMs to learn the rate of degradation in accuracy for each author – an approach that was found to be much more accurate than a baseline of one-class SVM.

Similarly to other domains, ensemble methods have been found to be effective for authorship analysis. For example, Kourtis and Stamatatos (2011) employed a semi-supervised co-training approach where unlabelled samples are iteratively labelled and added to the training set based on consensus between two base classifiers: an SVM trained on individual documents and a distance-based method trained on the concatenation of all the documents by each author. Another example is Kern et al.'s (2011) ensemble that combines the outputs of the base classifiers (logistic regression and bagging with random forests trained on different features) in a weighted voting scheme that considers precision and recall based on cross validation of the training set – only base classifiers with precision that exceeds a pre-configured threshold on a certain author are allowed to vote *for* this author, and similarly, only base classifiers with above-threshold author recall are allowed to vote *against* this author.

Our contribution to the authorship analysis field is in the application of the topical user models described in Chapter 4 to authorship attribution. We know of

only one previous case where topic models were used for authorship attribution: Rajkumar et al. (2009) reported preliminary results on using LDA topic distributions as feature vectors for SVMs, but they did not compare the results obtained with LDA-based SVMs to those obtained with SVMs trained on tokens only (we present the results of such a comparison in Section 5.3). We know of two related studies that followed the publication of our initial LDA-based results in (Seroussi et al., 2011c): Wong et al.'s (2011) work on native language identification with LDA, and Pearl and Steyvers's (2012) study of authorship verification where some of the features are topic distributions. While Wong et al. (2011) reported only limited success (perhaps because an author's native language may manifest itself in only a few words, or maybe due to dataset-specific issues), Pearl and Steyvers (2012) found that topical representations helped them achieve state-of-the-art verification accuracy. Pearl and Steyvers's findings further strengthen our hypothesis that topic models yield meaningful author representations. We take this observation one step further by defining our DADT model and applying it to several authorship attribution scenarios, where it yields better performance than LDA-based approaches and methods based on the AT model (Section 5.3). Our DADT-based methods can potentially be applied to verification and open-set attribution in a similar way to Tanguy et al.'s (2011) thresholding approach (described above), and can also be used as part of classifier ensembles. In addition, given the similarity of authorship profiling problems to authorship attribution, it is likely that DADT can be successfully adjusted to handle profiling scenarios. Such extensions are left for future work (Section 8.2).

2.4 Sentiment Analysis

Sentiment analysis (or opinion mining) deals with inferring people's sentiments and opinions from texts (Pang and Lee, 2008; Liu and Zhang, 2012). This area has received considerable attention in recent years due to the large amounts of user-generated texts available online and the many applications that are enabled by the ability to extract sentiments from texts. For example, Metavana (`www.metavana.com`) and Attensity (`www.attensity.com`) provide companies with sentiment analysis tools to detect public opinion about their products as expressed in social media; SocialMention (`www.socialmention.com`) is a search engine that presents a summary of the sentiment expressed towards the searched keywords alongside traditional search results; and RankSpeed (`www.rankspeed.com`) is a search engine that allows users to specify the sentiment that they are looking for together with content keywords. Many sentiment analysis tasks have been explored, including: polarity inference, which aims to infer the overall positivity of a given text; subjectivity classification, where the goal is to separate subjective segments from objective segments;

perspective identification, which deals with identifying the underlying perspective expressed in opinionated texts (e.g., the author's political point of view); and the identification of opinions towards specific entities and aspects in, e.g., product reviews (Pang and Lee, 2008).

Our contribution to the sentiment analysis field is in introducing a user-aware approach to polarity inference (Chapter 6). Our work is motivated by the insight that user identity plays a role in the way users express their sentiments. While this link between user identity and language use has been recognised by other researchers (Section 2.4.2), most of the work in the field does not directly harness user identity to improve the accuracy of sentiment analysis methods. We show that considering user identity can help improve polarity inference accuracy, as our framework, which combines the outputs of user-specific inference models, outperforms two baselines: one that ignores authorship information, and another that considers only the model learned for the user who wrote the text whose polarity we want to infer.

In this section, we focus on surveying previous research on the key task of polarity inference (Section 2.4.1), and then discuss the details of several user-aware studies of sentiment analysis (Section 2.4.2).

2.4.1 Polarity Inference

Polarity inference is one of the most commonly-attempted tasks in the sentiment analysis field (Pang and Lee, 2008; Liu and Zhang, 2012). In the binary case, it consists of classifying text as conveying either positive or negative sentiment. In the multi-way case, which has garnered less attention, the goal is to determine the polarity of texts on a scale of more than two values. Some researchers treat the multi-way problem as a classification problem (i.e., where texts are to be classified into three or more polarity categories), while others see it as a regression problem (i.e., where texts are to be assigned a real-valued score). We group both variants under the name "inference", as similar techniques and feature types are often applicable to either problem.

Binary polarity inference has been an active research area since the early days of sentiment analysis (Morinaga et al., 2002; Pang et al., 2002; Turney, 2002). The two main approaches are unsupervised inference, which often relies on external knowledge sources (e.g., a lexicon or the web), and supervised inference, which requires training texts that are labelled for polarity. The seminal studies by Turney (2002) and Pang et al. (2002) respectively serve as prime examples of these two approaches.

Turney (2002) studied binary classification of reviews from several domains (e.g., movies and cars). He employed an unsupervised approach where the semantic orientation of phrases is classified as either positive or negative according to their appearance in proximity to the words "excellent" or "poor" when performing a web

search for the given phrases. The overall polarity of the review is then determined by the average semantic orientation of the phrases in the review. In Turney's experiments, the accuracy of this approach varied from 66% for movie reviews to 84% for car reviews.

Pang et al. (2002) worked on classification of movie reviews as either positive or negative. They employed a supervised approach, comparing three different classification algorithms: naive Bayes, maximum entropy, and SVMs. They found that SVMs yielded the best performance of the classification algorithms, and that using unigram presence is superior to using other feature types such as unigram and bigram frequency. In addition, they found that appending part-of-speech tags to unigrams and considering only adjectives in isolation was of little benefit in terms of classification accuracy. The accuracy of their approach was 83% for the best method, which is much higher than what Turney (2002) achieved for movie reviews (though different datasets were used).

Since these early studies, there has been much work on polarity inference, exploring various domains, text granularities (e.g., sentences and whole documents), and feature types. Interest in polarity inference now extends beyond its origins in natural language processing, e.g., it is sometimes used as a benchmark task for new machine learning algorithms (Blei and McAuliffe, 2007; Mao and Lebanon, 2009; Zhu and Xing, 2010). Recently, Pang and Lee (2008), Liu (2010) and Liu and Zhang (2012) published surveys of sentiment analysis research, which provide more detailed information about polarity inference and other sentiment-related tasks. Examples of recent work on binary polarity inference include Dasgupta and Ng's (2009) study of semi-supervised classification of movie and product reviews, Paltoglou and Thelwall's (2010) comparison of feature weighting schemes for binary classification on several datasets, and Lu et al.'s (2011a) work on joint classification of texts in several languages.

An early example of *multi-way polarity inference* is by Yu and Hatzivassiloglou (2003), who studied three-way unsupervised classification of journal articles, where a category of *neutral* texts was considered in addition to the positive and negative categories (note that "neutral" does not necessarily mean "objective", as an opinionated text can carry a neutral opinion). Another example is Gamon's (2004b) dataset of customer feedback, where satisfaction is measured on a four-point scale. Rather than classifying the texts into four categories, Gamon reduced the problem to two binary classification problems: (1) texts rated 1 versus texts rated 4, and (2) 1 and 2 versus 3 and 4. He used SVMs to address these problems and found that better performance was obtained on the first problem (as expected). One of the earliest attempts at full-blown polarity rating inference was made by

Pang and Lee (2005), who experimented with both supervised classification and regression of movie reviews, with the former performing better for a 3-star scale and the latter achieving better results in the 4-star case. Since then, many multi-way inference studies have been performed, e.g., Snyder and Barzilay (2007) and Sauper et al. (2011) inferred ratings from restaurant reviews with models that consider several aspects of the restaurant-going experience, and Blei and McAuliffe (2007) and Zhu and Xing (2010) developed topic modelling techniques to address general text-based regression problems, which they tested on Pang and Lee's (2005) movie review dataset.

Polarity inference results vary depending on the method, domain, text length and other dataset-specific properties. Unsurprisingly, multi-way polarity inference results are often inferior to binary inference results. In Chapter 6 we focus on multi-way polarity inference as an example of a sentiment analysis problem. Our main hypothesis is that a user-aware analysis would be especially beneficial in this case because ratings on a non-binary scale are more open to interpretation than binary ratings (e.g., the difference between a rating of 6 and 7 on a 10-point scale is not as clear-cut as the difference between positive and negative), and thus every user has a different "feel" for the rating scale. Our experimental results support this hypothesis, since we found that our user-aware approach yields improved performance in comparison to the two baselines we considered (one that ignores authorship information, and another that considers only the model learned for the user who wrote the text whose polarity we want to infer).

2.4.2 User-aware Sentiment Analysis

Several researchers found that authorship affects sentiment analysis performance (Pang and Lee, 2005; Lin et al., 2006; Greene and Resnik, 2009; Mao and Lebanon, 2006). However, to the best of our knowledge, there has not been any work that utilises the link between authorship and sentiment to improve performance in settings where the only thing given – apart from the texts – is the identity of the authors. Our main contribution to the sentiment analysis field is in providing empirical evidence for the connection between users and the sentiments expressed in their texts, and in introducing a polarity inference approach that successfully harnesses this connection to improve performance (Chapter 6). In this section, we review several studies that used this connection or at least acknowledged its existence (in contrast to the majority of studies, which do not take authors into account).

Pang and Lee (2005) performed multi-way polarity inference on a dataset of movie reviews by four different authors. They reported results obtained by *separately* training and testing their methods on each author's reviews to avoid having to deal with cross-author differences. Their focus was on improving the performance

of generic polarity inferrers by considering similarity between texts as measured according to the percentage of positive sentences. Even though Pang and Lee did not attempt to harness authorship information in their work, their insight that authorship affects performance is a motivating factor of our work.

Lin et al. (2006) and Greene and Resnik (2009) found that authorship affects performance in the perspective identification task. They tested their methods on the Bitter Lemons dataset, which contains pro-Palestinian and pro-Israeli articles, half of them written by two editors and the other half by various guest writers. They found that training and testing on articles by the editors resulted in near-perfect accuracy, while training and testing on articles by the guest writers yielded lower accuracy. In addition, training on articles by the guest writers and testing on the editors' articles resulted in higher accuracy than the reverse case. Like Pang and Lee (2005), neither Lin et al. (2006) nor Greene and Resnik (2009) harnessed authorship information in their methods. However, their findings indicate that author-awareness could be of benefit in tasks beyond polarity inference.

An example of an early study that harnesses user identity is by Mao and Lebanon (2006), who developed a model that measures changes in sentiment from sentence to sentence (which they named the *sentiment flow*), and utilises these changes to infer the overall polarity of documents. Most of Mao and Lebanon's evaluation focused on a version of their model that does not consider authors. However, they hypothesised that sentiment flow would vary across *review* authors since, e.g., some may first list pros and then cons, while others may discuss different aspects of the topic under review regardless of their sentiment towards these aspects. Mao and Lebanon presented some evidence supporting this hypothesis by applying an author-aware version of their model to a dataset of reviews by two authors. The results showed that different authors exhibit different sentiment flow patterns, but accuracy results obtained with the author-aware model were not presented. As our focus is on improving the overall accuracy of polarity inference methods, our evaluation datasets include texts by many authors, and we test the performance of our approach with different numbers of training texts per author.

Several researchers considered users in conjunction with other information beyond author names. For example, Li et al. (2011) developed a tensor factorisation technique to infer the polarity of product reviews in a scenario where it is assumed that both the authors and the items under review are known. Unsurprisingly, they found that their approach outperforms baselines that either consider only the reviews and their polarities, or only the users, items and polarities (but not the texts). An example of integrating a different type of information is by Tan et al. (2011), who employed social network connections in their work on determining the sentiments

of Twitter users towards public figures, sport teams and news corporations. In addition, Pang and Lee (2008) list several studies that consider reviewer credibility when determining review usefulness. Our work differs from these lines of research in that we assume that the only information available to our algorithms apart from the texts is the identity of the authors.

2.5 Recommender Systems

Recommender systems help users deal with information overload by finding and recommending items of personal interest (Resnick and Varian, 1997). While interest in recommender systems has been high since the 1990s, in recent years recommender systems have become more ubiquitous and are used in various domains. Examples include Amazon's product recommendations (`www.amazon.com`), Facebook's friend suggestions (`www.facebook.com`), and Google Play's Android application recommendations (`play.google.com`). Scalable implementations of recommendation algorithms are freely available through projects like Apache Mahout, which is deployed in many leading websites.[4] Academic research on recommender systems has also seen a steady growth, evidenced by the organisation of several workshops (e.g., Cantador et al., 2010; Anand et al., 2011; Degemmis et al., 2011) and a conference series dedicated to this topic (`recsys.acm.org`), along with the publication of several surveys (e.g., Burke, 2002; Adomavicius and Tuzhilin, 2005; Schafer et al., 2007; Su and Khoshgoftaar, 2009) and books (e.g., Jannach et al., 2010; Ricci et al., 2011).

Two key considerations in building recommender systems are the choice of an algorithm for recommendation generation, and the design of user interfaces to display recommendations and obtain user input (Herlocker et al., 2004). Choosing an algorithm for recommendation generation is partly dependent on the type of data available to the system. For example, when past ratings by the users are available, it is possible to employ *collaborative rating prediction* to predict the rating a given user would assign to a given item (Koren and Bell, 2011). The predicted ratings can then be employed to generate a personalised ranked list of recommended items. Other recommendation generation paradigms include *content-based* recommendation, which relies on domain-specific knowledge about the content of the items (Lops et al., 2011); *context-aware* recommendation, where recommendations are generated with regards to contextual features such as the user's current location (Adomavicius and Tuzhilin, 2011); and *constraint-based* recommendation, where the system can elicit specific constraints regarding the items that are to be recommended, e.g., when recommending cars, price is likely to be an important factor for most users (Felfernig et al., 2011).

[4] According to `cwiki.apache.org/MAHOUT/powered-by-mahout.html`, the Mahout recommendation engine is used by AOL (`www.aol.com`), Foursquare (`www.foursquare.com`), and Mendeley (`www.mendeley.com`), among others.

Our contribution to the field of recommender systems is in the area of collaborative rating prediction: we enhance the popular matrix factorisation algorithm to consider user information when generating predictions for new users, who submitted few ratings (Chapter 7). We show that either explicit demographic information or implicit text-based information (in the form of topical user models) can be employed to improve predictive accuracy for such users. In addition, neighbourhood-based approaches to collaborative rating prediction serve as inspiration for our user-aware approach to polarity inference (Chapter 6).

In this section, we focus on previous recommender system studies that are directly relevant to the study presented in this thesis (more thorough reviews of research in this field can be found in the books and surveys mentioned above). In Section 2.5.1, we discuss previous work on collaborative rating prediction. Recommenders that incorporate demographic information are surveyed in Section 2.5.2, and systems that consider user-generated texts are reviewed in Section 2.5.3.

2.5.1 Collaborative Rating Prediction

Under the *collaborative recommendation* (or *collaborative filtering*) approach, recommendations are generated mainly based on preferences by the user population, without requiring any domain-specific knowledge about the items (Goldberg et al., 1992; Balabanovic and Shoham, 1997; Adomavicius and Tuzhilin, 2005). Recommendation generation often requires a *rating prediction* algorithm, where the input consists of past preferences encoded as explicit or implicit ratings (which can be discrete or real-valued), and the output is a rating for each user-item pair. These predictions enable the personalised ranking of items according to each user's inferred tastes.

Collaborative rating prediction techniques are often categorised as being either *memory-based* or *model-based* (Breese et al., 1998). Memory-based methods find the most similar *training* users or items to the *target* user or item respectively (i.e., the neighbourhood of the target user or item), and base their rating predictions on past ratings from the neighbourhood. By contrast, model-based methods build a model from all the available ratings and base their predictions on the model, rather than directly on the ratings. While some model-based approaches have been shown to outperform memory-based methods in terms of predictive accuracy and prediction generation time (e.g., Bohnert et al., 2009; Koren et al., 2009), memory-based methods still have their merits (Desrosiers and Karypis, 2011), e.g., their output is often easy to explain to users (Tintarev and Masthoff, 2011). Further, it has been shown that combining predictions made by model-based algorithms with predictions produced by memory-based methods yields higher accuracy than that obtained by either algorithm in isolation (Jahrer et al., 2010). Our user-aware approach to polarity inference is inspired by memory-based collaborative methods (Chapter 6), while

our contribution to the field of recommender systems is in extending the popular matrix factorisation algorithm for rating prediction (which is model-based) to consider user information when generating predictions for new users (Chapter 7). Hence, we survey previous work from both lines of research in this section.

Memory-based Approaches

Memory-based approaches have been in use since the earliest recommender systems, originally motivated by the idea that similar users have similar tastes (Goldberg et al., 1992; Breese et al., 1998). This idea was implemented in the Tapestry system, where users manually specified which other users should be employed for generating recommendations (Goldberg et al., 1992). Later systems, such as GroupLens (Resnick et al., 1994; Konstan et al., 1997), automatically detected similar users by calculating the target user's similarity to the training users, and predicted the rating the target user would give to a target item by combining the ratings given to this item by the most similar training users – this combination was done in a weighted manner, according to each training user's similarity to the target user. A variant of this idea, introduced by Sarwar et al. (2001), is to calculate item-to-item similarity and base predictions on the target item's similarity to items that the target user has already rated. These two approaches were combined by Wang et al. (2006) in an algorithm that bases predictions on a combination of: (1) ratings of the target item by users similar to the target user; (2) ratings by the target user of items similar to the target item; and (3) ratings by users similar to the target user of items similar to the target item.

Several aspects should be taken into consideration when employing memory-based methods, including (Herlocker et al., 1999; Desrosiers and Karypis, 2011):

- Choosing a *similarity measure*. For example, in user-based variants either cosine similarity or Pearson's correlation coefficient are commonly used to compare pairs of users based on the ratings given to co-rated items (i.e., items that were rated by both users) (Adomavicius and Tuzhilin, 2005; Desrosiers and Karypis, 2011). Many other similarity measures have been explored, including measures based on the mutual information of ratings (Brun et al., 2009), and on extraneous information about the users (Section 2.5.2).
- Setting the *size of the neighbourhood*. This affects predictive accuracy, as well as the time it takes to generate predictions (Herlocker et al., 1999; Sarwar et al., 2001).
- Selecting a *rating aggregation approach*. This also impacts accuracy, e.g., Herlocker et al. (1999) found that normalising the ratings and using the target user's mean as a base predictor reduced the predictive error by more than 30% over using a simple weighted average.

Other considerations include weighting users and items according to the number of available ratings (Herlocker et al., 1999), accounting for "expert" users (Amatriain et al., 2009) or "mentors and leaders" (Brun et al., 2011), and whether to cache similarities and how often they should be recalculated (Owen et al., 2011).

The advantages of memory-based approaches include their simplicity, ease of implementation (Desrosiers and Karypis, 2011), and that their predictions can often be easily explained to users (Tintarev and Masthoff, 2011). Nonetheless, memory-based approaches have some limitations. One limitation is their susceptibility to the new user and new item problems, where poor accuracy is obtained for target users and items with few ratings (Adomavicius and Tuzhilin, 2005). Another limitation arises in deployed systems, where special care is required to enable processing of datasets containing millions of users and items, e.g., if the neighbourhood size is not set to a reasonable number, the calculation of predictions can become computationally prohibitive (Sarwar et al., 2001).

Model-based Approaches

In parallel to the development of memory-based approaches, model-based rating prediction has also received considerable attention (Adomavicius and Tuzhilin, 2005). Early examples include works by Ungar and Foster (1998), who based predictions on probabilistic clustering of users and items, and by Breese et al. (1998), who experimented with a cluster model of users and with Bayesian networks. Other models that have been shown to be effective are Hofmann's (2003) extension of probabilistic latent semantic analysis to handle continuous rating data, Bohnert et al.'s (2009) model that employs spatial processes to predict interests of museum visitors, and Harvey et al.'s (2011) latent variable model, which can be seen as a Bayesian version of Koren et al.'s (2009) matrix factorisation approach.

Matrix factorisation (MF) is a model-based technique that played a key part in the approach of the team that won the million dollar Netflix Prize competition, where over the course of three years participating teams developed algorithms to improve the accuracy of Netflix's baseline by 10% (Koren et al., 2009). The main idea behind matrix factorisation is to decompose the user-item rating matrix to uncover latent factors that can be seen as representing interactions between user interests and item characteristics. Matrix factorisation methods can be extended to include information beyond ratings, e.g., rating timestamps (Koren, 2010), social network information (Jamali and Ester, 2010), and item metadata (Dror et al., 2011). We describe the basic matrix factorisation algorithm in detail in Section 7.1, and extend it to consider user information in Section 7.2.

While using matrix factorisation for rating prediction has considerable advantages in terms of accuracy and runtime, it still tends to perform poorly on new users, as we demonstrate empirically in Section 7.4.2. Performance on new users

has received less attention than overall accuracy in many studies. For example, in the Netflix competition, the prize was awarded to the team that obtained the lowest error when measured over all the test ratings (Koren et al., 2009). This means that equal weight is given to, e.g., errors incurred for a user with 100 training ratings and 100 test ratings, and errors incurred for 100 users with one training rating and one test rating each. Our extension to matrix factorisation is specifically geared towards new users, and thus our evaluation is focused on users who submitted no ratings at all or exactly one rating (Section 7.4.1).

2.5.2 Demographic Recommenders

Demographic information has been considered in the recommendation generation process in several previous studies. One early example is Lifestyle Finder, which used explicit demographic and lifestyle information to categorise users into one of 62 pre-defined clusters, and delivered recommendations accordingly (Krulwich, 1997). In contrast to Lifestyle Finder, Pazzani's (1999) system extracted demographic information from user websites and used it as features for a binary classifier that was trained for each item, where the class labels were positive or negative ratings. The performance of this method was rather poor compared to other approaches tested in that paper, but Pazzani found that using the classifier's predictions in an ensemble setting had a positive impact on recommendation precision. Note that this early study was done on a rather small dataset with only 58 items, and that training a classifier for each item may become too computationally expensive in settings with many more items.

More recent studies were conducted by Lekakos and Giaglis (2007), Vozalis and Margaritis (2007), Gong (2009) and Hu and Pu (2011), who introduced extensions to memory-based collaborative rating prediction that consider information about the users. Specifically, Lekakos and Giaglis (2007) used demographics and lifestyle indicators to calculate similarities between users and found that it improves accuracy for new users. By contrast, Vozalis and Margaritis (2007) used only demographics to measure user similarity, but found that it does not improve performance. Gong (2009) took Vozalis and Margaritis's (2007) work one step further by employing demographics to measure similarity and to populate the rating matrix before generating predictions, which resulted in reduced predictive error (rating matrix population was also explored by Lekakos and Giaglis, who similarly found that it has a positive effect on performance). Hu and Pu (2011) followed a similar approach to Gong's (2009), but applied it to user personality traits rather than to demographics. Like Gong, Hu and Pu found that their method outperforms traditional memory-based prediction when applied to new users. Our approach builds on these results in that we also employ demographic information to improve accuracy.

The key difference is that rather than taking a memory-based approach, we extend the matrix factorisation algorithm, which has been shown to be more accurate than memory-based methods (Koren et al., 2009) – a result that we reproduced on our datasets in preliminary experiments.

The most similar work to the algorithm we present in Chapter 7 is probably by Koren et al. (2009), who suggested a way of considering demographic user attributes in matrix factorisation. To the best of our knowledge, Koren et al. did not evaluate their suggested method, perhaps because their focus was on the Netflix dataset, which does not contain demographic information. As our focus is on new users, our algorithm differs from Koren et al.'s (2009) suggestion in several ways, which are discussed in Section 7.2 (e.g., we employ a switching approach to specifically target new users, and enable the use of partial demographic information by allowing probabilistic assignment of attributes). Two other related studies are by Gantner et al. (2010) and Shan and Banerjee (2010), who developed algorithms for considering *item* information in matrix factorisation. These studies were performed in parallel to our rating prediction work, and were published at around the same time when we submitted our work for publication in (Seroussi et al., 2011a). Like Koren et al. (2009), both Gantner et al. and Shan and Banerjee noted that their algorithms can be extended to consider user information, but did not empirically test these claims. Implementing these extensions and comparing their performance to the performance of our approach is left for future work.

2.5.3 Text-aware Recommenders

The main disadvantage of using demographics, lifestyle and personality indicators in rating prediction is that in general they need to be explicitly obtained from the users. Our focus in Chapter 7 is on new users who have supplied few explicit ratings. Such users are unlikely to divulge personal information and take the time to complete surveys about their lifestyle preferences and personality. Therefore, we propose to obtain information about users from texts they write separately from giving explicit ratings, such as message board posts. Our approach is based on the finding that texts are implicitly indicative of user characteristics (e.g., demographics and personality traits, as discussed in Section 2.3), and often communicate user interests (either implicitly or explicitly). We harness this information in our extension to the basic matrix factorisation algorithm, where we employ topical user models to obtain user attributes.[5]

[5]As discussed in Section 7.3.3, we use topical user models rather than inferring demographics and personality traits, because accurate inference often requires domain-specific labelled data, and we found that topical user models capture authorship traits (Chapter 5), which are indicative of demographics and personality (Section 2.3).

We are aware of several other attempts to utilise user texts in recommender systems (e.g., Aciar et al., 2006; Leung et al., 2006; Jakob et al., 2009; Ganu et al., 2009). However, they all focus on reviews rather than on more general texts, which makes them different from the work we present in Chapter 7.

One of the earliest attempts at incorporating texts into recommender systems was made by Aciar et al. (2006, 2007), who aggregated opinions from product reviews and displayed a product score based on queries by the target users. In their system, nothing is known about the target users apart from their queries, which explicitly specify their preferences. Thus, there is no collaborative rating prediction done in their case, which makes it very different from our work: we focus on collaborative rating prediction, while striving to minimise the amount of explicit information that the users need to supply about themselves and their preferences.

Leung et al. (2006) aimed to integrate sentiment analysis and collaborative recommendation, but their approach was sequential rather than integrative. They first inferred the polarities of user reviews and then used these polarities as input for the rating prediction algorithm. Thus, the texts were not directly integrated into the prediction algorithm, which was still based only on ratings.

In contrast to Leung et al. (2006), Jakob et al. (2009) and Ganu et al. (2009) performed a more in-depth analysis of the review texts. In both cases it was shown that it is possible to improve the accuracy of rating predictions by using polarity inference to obtain the ratings of item aspects from review texts, and using these ratings as additional features for rating prediction algorithms. More recent review-based studies can be seen as extensions of the work done by Jakob et al. and Ganu et al.. Examples include: Zhang et al.'s (2010) approach, where unsupervised polarity inference was employed instead of supervised inference; Esparza et al.'s (2011) work, which focused on evaluation measures that go beyond predictive accuracy (e.g., recommendation novelty); Moshfeghi et al.'s (2011) study, which explored fine-grained sentence-level emotions rather than focusing only on polarity; and Ganu et al.'s (2012) extension to their earlier work, which includes an improved rating prediction model and stronger baselines than those used in (Ganu et al., 2009). All these studies are different from our approach, as we focus on new users who have not necessarily written reviews, and we use texts to model the users directly, rather than the relationship between individual users and item aspects.

2.6 Summary

In this chapter, we surveyed the fields in which our research lies, focusing on the specific areas that serve as the backdrop of this thesis: topic modelling, authorship analysis, sentiment analysis and recommender systems. We showed that while the roots of these research areas are somewhat disparate, they share several common

themes, such as the use of machine learning and statistical techniques, and the existence of users (which often author texts) either as an integral part of the tasks (in authorship attribution and rating prediction), or as a less central feature (as text authors – which are often ignored – in topic modelling and polarity inference). We hope that the work presented in this thesis will help bridge some of the gaps between these areas. The benefits of addressing these gaps are demonstrated by our empirical results, which show the utility of using topic modelling techniques for authorship attribution (Chapter 5), considering authors when performing polarity inference (Chapter 6), and employing topical user models to deliver more accurate rating predictions for new users (Chapter 7).

Chapter 3

Methodology and Data

This chapter outlines the experimental approach employed in this thesis, and describes the datasets used to evaluate our methods. Broadly speaking, our approach is data-driven and empirical, utilising real-life data samples to ensure that our methods are applicable to realistic situations. Section 3.1 describes the experimental setup we employed in this study, and Section 3.2 discusses our choice of evaluation criteria. The datasets used in our experiments are introduced in Section 3.3, and Section 3.4 presents the steps we took to preprocess the data, and the external tools we used.

3.1 Experimental Setup

We employed the same experimental setup for the three main tasks considered in this thesis: authorship attribution, polarity inference, and rating prediction. In all cases, we assumed a supervised learning setup, where the methods are given *labelled training samples* in advance, which they use for model building. Then, given an *unlabelled test sample*, the methods use the model built from the training samples to assign a label to the test sample. For example, for authorship attribution, the training samples are texts and their labels are authors, while the test samples are only the texts (i.e., the real author is kept hidden from the authorship attribution method).

In our experiments, we employ ten-fold cross validation, using stratified sampling where possible. Stratified ten-fold cross validation is a procedure that is commonly used to evaluate performance based on samples with known labels (Witten and Frank, 2005). Under this procedure, the samples are split into ten distinct folds (or subsets), such that the samples from nine folds are used for training the models, and the samples from the remaining fold are used for testing. The folds are sampled in a stratified way that ensures that the label balance is the same across all folds. Stratified sampling is not possible in cases where there are less than ten samples for each label, in which case we use random sampling. The measure used for evaluation

is calculated based on the labels assigned by the tested methods to the test samples, as described in Section 3.2. To increase the reliability of the measures, we repeat the cross validation procedure five times, using five different random seeds that yield different fold splits, as advocated by Witten and Frank (2005). Hence, the reported results are averages obtained by running $10 \times 5 = 50$ folds overall.

While ten-fold cross validation gives a good idea about the performance of the methods on datasets with similar properties to the dataset used for cross validation, it does not always sufficiently test other interesting scenarios. Specifically, in some cases we are interested in the performance of our methods on particular types of users. For example, in our rating prediction experiments we are interested in users who submitted few ratings (Chapter 7). As ten-fold cross validation is performed over the samples, the target measure is averaged across all test samples for each fold. In the case of rating prediction, the samples are ratings by users to items. Thus, users with many training and test ratings will affect the performance measure more than users with few ratings, thereby leading us to favour methods that do not necessarily perform well on users with few ratings.

To address this issue, we employ the GivenX protocol, where each target user has exactly X training samples (Breese et al., 1998). Specifically, we perform ten-fold cross validation over *users*, where we split the users into ten folds and iterate over the folds, using nine folds as the training folds and the remaining fold as the test fold. The model is trained on all the samples by the users in the training folds, and exactly X samples by each target user in the test fold. The model is then tested on the remaining samples by each target user. We repeat this process five times with different random seeds, as we do in ten-fold cross validation over samples. Employing the GivenX protocol is more costly in terms of runtime than ten-fold cross validation over samples, because it requires running a separate experiment for each value of X. However, it enables us to gain insights into the performance of our methods on specific user types. It is also worth noting that we do not use the GivenX protocol to compare the performance of the same method across different X values (e.g., testing how the performance of method A varies from Given1 to Given100), because the test samples vary across X values. Rather, we use this protocol to compare different methods under the same conditions, e.g., by comparing the performance of method A to that of method B under the Given1 scenario.

3.2 Evaluation Criteria

Our evaluation criteria are task-dependent: For authorship attribution, which is a classification task, we employ the *accuracy* measure; and for polarity inference and rating prediction, which are regression tasks, we use *root mean squared error* (RMSE). Both measures are calculated based on the test samples in each fold,

and then averaged across all the folds. Specifically, we denote the set of test samples in a given fold by $\mathcal{T}$ and calculate these measures as follows.

Accuracy. For authorship attribution, $\mathcal{T}$ contains documents with known authors, whose identity is withheld from the attribution methods. We denote the actual author of document d with a_d, and the author returned by the authorship attribution algorithm with $\hat{a}_d$. The accuracy of the method is the percentage of documents that were correctly assigned to their actual authors:

$$\text{Accuracy} = \frac{\sum_{d \in \mathcal{T}} I(a_d = \hat{a}_d)}{|\mathcal{T}|} \tag{3.1}$$

where I is the indicator function that is equal to 1 if its argument is true, and 0 otherwise.

We chose the accuracy measure for authorship attribution because it is easy to interpret, as it summarises the performance of classification methods with a single value (Witten and Frank, 2005).

RMSE. For polarity inference, $\mathcal{T}$ contains sentiment-bearing documents with known polarities, which are withheld from the inference methods. We denote a sentiment-bearing document with q to differentiate it from a document d that is not known to bear sentiment. Given a sentiment-bearing document q that was written by user u, we denote its polarity with r_{uq}, and the inferred polarity with $\hat{r}_{uq}$. The RMSE is then:

$$\text{RMSE} = \sqrt{\frac{\sum_{r_{uq} \in \mathcal{T}} (r_{uq} - \hat{r}_{uq})^2}{|\mathcal{T}|}} \tag{3.2}$$

Similarly, the RMSE for rating prediction is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{(u,i) \in \mathcal{T}} (r_{ui} - \hat{r}_{ui})^2}{|\mathcal{T}|}} \tag{3.3}$$

where $\mathcal{T}$ now contains user-item pairs (u, i). The rating prediction method is asked to predict the rating $\hat{r}_{ui}$ that user u would give to item i, while the actual rating r_{ui} is withheld from the rating prediction method.

The RMSE measure is commonly used to evaluate regression methods, such as rating prediction (Koren et al., 2009). Another popular measure is *mean absolute error* (MAE), which is correlated to RMSE, but does not penalise large errors as much as RMSE (Witten and Frank, 2005). We chose to report only the RMSE (as opposed to reporting, e.g., both MAE and RMSE) in order to keep the results easy to read and interpret, as it is often the case that methods that perform well according to the RMSE measure also perform well according to other measures, such as MAE (Witten and Frank, 2005).

Dataset	Description	Used in Chapter 5	6	7
IMDb62	Rated movie reviews and message board posts by 62 prolific users	X	X	
IMDb1M	Rated movie reviews and message board posts by a million random users	X	X	X
Judgement	Judgements by three judges	X		
PAN'11	Emails by 72 authors	X		
Blog	Blog posts by 19,320 authors	X		
MovieLens	Ratings by 6,040 users with demographic information			X

Table 3.1: Dataset outline

Statistical Significance. Statistically significant differences in accuracy or RMSE are reported when $p < 0.05$ according to a paired two-tailed t-test, which is performed as described in (Witten and Frank, 2005). Throughout this thesis, we either explicitly state the results of the relevant t-tests in the text, or, when reporting results in a table, we highlight the best results in boldface, with several highlighted entries meaning that the differences between the top results are not statistically significant.

3.3 Datasets

We experimented with six datasets: IMDb62, IMDb1M, Judgement, PAN'11, Blog and MovieLens. The first three datasets were collected and introduced by us, and are freely available for research use.[1] The other datasets were introduced by other researchers, are publicly available, and were used to facilitate comparison between our methods and previous work. Due to the nature of the tasks considered in this study, not all the datasets could be used for all the tasks. Table 3.1 provides brief descriptions of the datasets and specifies the chapters in which they are used. We discuss each dataset in detail in the following sections.

3.3.1 IMDb62

Our main goal in creating the IMDb62 dataset was to enable testing of our author-aware polarity inference approach (Chapter 6). Existing datasets were unsuitable for this purpose. For example, the popular *Sentiment Scale Dataset* (Pang and Lee,

[1] IMDb62 and IMDb1M are available upon request. The Judgement dataset can be downloaded from `www.csse.monash.edu.au/research/umnl/data`.

2005) includes movie reviews by four different users and mappings to 3-star and 4-star rating scales. We could not use this dataset, as this small number of users is inadequate to support experiments regarding the impact of authorship on sentiment. Another example is Jindal and Liu's (2008) dataset that contains product reviews by many users but no objective texts, which we need to test the effect of modelling users based on texts that they write independently of reviewing items (objective texts are also not included in Pang and Lee's dataset).

We created the IMDb62 dataset in May 2009 by crawling the *Internet Movie Database* (IMDb) (`www.imdb.com`). We downloaded all the movie reviews and message board posts written by prolific users, which are listed in `www.freewebs.com/bobthemoo`. Some reviews are associated with an overall rating, which is selected by the review's author from a 10-star scale. IMDb message boards are mostly movie-related, but some are about television, music and other topics. The following steps were performed in the creation of this dataset:

1. To reduce ambiguity, we ensured that each item is reviewed only once by each user. We did this by discarding multiple reviews for the same item by the same user.
2. Reviews without ratings were excluded from the dataset.
3. We excluded users who had less than 1,000 rated reviews. For each of the remaining users, we retained 1,000 rated reviews from the full set of reviews by using proportional sampling without replacement (i.e., the set of sampled reviews has the same rating distribution as that of the complete set of reviews for each user).
4. Explicit ratings were automatically filtered out from the review texts by matching regular expressions, e.g., "5/10" was removed from texts such as "this movie deserves 5/10".
5. For each user, all the message board posts are included (no sub-sampling of message board posts was performed). Some users have not submitted any posts, while others wrote hundreds to thousands of posts.

Table 3.2 presents some statistics of the IMDb62 Dataset.[2] The lower overall number of message board posts, together with the fact that posts are shorter than reviews on average, means that modelling IMDb62 users based on their posts would be more challenging than using both posts and reviews. Nonetheless, our main goal in producing this dataset was to enable testing of author-aware polarity inference, which is why we chose to focus on users with many reviews. It is worth noting

[2]The token statistics were obtained by employing the tokenisation procedure described in Section 3.4. The number of posts per user excludes the 11 users who did not submit any posts.

Property	Value
Users	62
Rated reviews	62,000
Reviews per user	1,000
Tokens per review mean (stddev)	338.7 (223.6)
Posts	17,550
Users without posts	11
Posts per user mean (stddev)	344.1 (743.3)
Tokens per post mean (stddev)	80.7 (146.5)

Table 3.2: IMDb62 statistics

that in the evaluation of our polarity inference approach we use the GivenX protocol (Section 3.1), so we do not always assume that every user has submitted 1,000 reviews. In fact, we show that our methods yield improved performance even when the number of given reviews is small (Section 6.5).

An observation that motivated our polarity inference approach is that different users may have different interpretations of the rating scale, e.g., two users may express seemingly similar opinions, but assign different ratings to their opinions (Chapter 6). Further, users select the items they review, and therefore they may choose to submit only reviews with extreme ratings (or more generally, some people tend to be vocal only about things they feel strongly about, while others will form and voice an opinion about anything and everything). This is exemplified in IMDb62, which displays a large variability of rating distributions. For example, some users have more than 40% 10-star ratings and almost no 1–4 star ratings, while others have most of their ratings in the 1–5 star range.

As noted above, we mainly use IMDb62 in polarity inference experiments (Chapter 6). However, we also use it in authorship attribution experiments as an example of a dataset of informal texts by prolific authors (Chapter 5). We do not use IMDb62 in our rating prediction experiments (Chapter 7) because it contains only 62 users. This is a somewhat unnatural setting for rating prediction, which often forms the basis of recommendations for many more users (Section 2.5). To test our rating prediction approach under a more natural setting, we collected another IMDb dataset, which is described in the next section.

3.3.2 IMDb1M

While the IMDb62 dataset is useful for testing our methods on small-to-medium scale problems (by using different subsets), it cannot be seen as an adequate representation of large-scale problems. This is especially relevant to the task of rating prediction, in which typical datasets contain thousands of users. For example,

the commonly-used MovieLens dataset contains about a million ratings by 6,040 users (Section 3.3.6), and the Netflix Prize dataset contains 100 million ratings by about 500,000 users (Koren et al., 2009). However, we do not know of any rating dataset that contains texts that are not directly related to the ratings (i.e., texts that are not reviews, such as message board posts). Hence, we created our own dataset, IMDb1M.

The IMDb1M dataset was crawled in July 2010 from IMDb by randomly generating one million valid IMDb user IDs and downloading the reviews and message board posts written by these users. Unfortunately, most of the randomly generated IDs led to users who submitted neither reviews nor posts – we found that about 5% of the entire user population submitted posts, while less than 3% wrote reviews.[3] After filtering out users who have not submitted any rated reviews and performing the same preprocessing steps as for IMDb62 (except for step 3 – sub-sampling of reviews), we were left with 22,116 users. These users, who make up the IMDb1M dataset, submitted 204,809 posts and 66,816 rated reviews. This is a suitable dataset for testing our hypothesis that utilising posts to model users is beneficial, as posts appear to be more prevalent than reviews.

It is worth noting that in the general case, one of our ultimate goals is to extract implicit and explicit information about users from texts they write, and employ it in tasks that require user information, e.g., when personalising user experience via recommendations (Section 2.5). An example of implicit information is demographic attributes inferred from texts (Section 2.3), and an example of relatively explicit information is user interests.[4] It may seem like 5% is a small percentage of the user population, which suggests that not many people would benefit from the personalisation yielded by the extracted information. However, we must remember that in many cases it is possible to obtain texts for much larger portions of the population, e.g., it is likely that more users communicate using emails and social media messages than those who write IMDb message board posts. Assuming that user consent is given, the techniques we introduce in this thesis can potentially be be applied to such texts, e.g., by recommender systems that are deployed as social media applications.

Table 3.3 presents some statistics of the IMDb1M Dataset. As for IMDb62 (Table 3.2), the number of posts per user excludes the 16,160 users who did not submit any posts. Similarly to IMDb62, IMDb1M reviews are much longer than posts on average. Since we use IMDb1M in our rating prediction experiments (Chapter 7), we included item statistics. The high sparsity of the user-item rating matrix (99.99%), together with the low average number of ratings (i.e., rated reviews) per user and

[3]Some of these users may have submitted ratings, but ratings without reviews are not publicly available.

[4]Interests are also often expressed implicitly, e.g., a fan of the horror film genre may write about many horror movies without explicitly mentioning the genre.

Property	Value
Users	22,116
Items	26,765
User-item rating matrix sparsity	99.99%
Rated reviews	66,816
Reviews per user mean (stddev)	3.0 (27.8)
Reviews per item mean (stddev)	2.5 (5.4)
Tokens per review mean (stddev)	270.5 (197.2)
Posts	204,809
Users without posts	16,160
Posts per user mean (stddev)	34.4 (163.4)
Tokens per post mean (stddev)	76.4 (121.9)

Table 3.3: IMDb1M statistics

per item, make rating prediction on this dataset challenging, even in comparison to popular datasets such as MovieLens, which is not as sparse (Section 3.3.6).

In addition to rating prediction experiments, we also use the IMDb1M dataset in authorship attribution and polarity inference experiments (Chapters 5 and 6 respectively). Its use can be seen as complementary to the IMDb62 dataset, as IMDb62 allows us to test scenarios in which the user population is made up of prolific users (though we can emulate non-prolific users by employing the GivenX protocol), while IMDb1M contains a more varied sample of the population.[5] However, since we did not impose a minimum threshold on the number of reviews or posts, the IMDb1M population is very challenging because it includes many users with few texts (e.g., about 71% of the users in IMDb1M wrote only one review).

3.3.3 Judgement

The Judgement dataset contains judgements by three judges who served on the Australian High Court from 1913 to 1975: Dixon, McTiernan and Rich. This dataset was created following rumours that Dixon ghost-wrote some of the judgements attributed to McTiernan and Rich.[6] We used standard authorship attribution methods

[5]Three users appear in both datasets. In IMDb62 these three users authored 3,000 reviews and 268 posts in total (about 4.8% of the total number of reviews and 1.5% of the posts), and in IMDb1M they authored 5,695 reviews and 358 posts (about 8.5% of the reviews and 0.2% of the posts). Note that the difference in the number of reviews is due to the sampling we performed when we created IMDb62, and the difference in the number of posts is due to the time difference between the creation of the two datasets.

[6]This dataset was created in collaboration with Professor Russell Smyth from the Department of Economics at Monash University.

to verify these rumours and to estimate the extent to which Dixon ghosted for McTiernan and Rich. The results of this work, together with a more detailed historical background, were reported in (Seroussi et al., 2011b).

The Judgement dataset is an example of a traditional authorship attribution dataset, as it contains only three authors who wrote relatively long texts in a formal language. In this thesis, we only use judgements with undisputed authorship, which were written in periods when only one of the three judges served on the High Court (Dixon's 1929–1964 judgements, McTiernan's 1965–1975 judgements, and Rich's 1913–1928 judgements). We removed numbers from the texts to ensure that dates cannot be used to discriminate between judges. We also removed quotes to ensure that the classifiers take into account only the actual authors' language use.[7] Employing this dataset in our experiments allows us to test our authorship attribution methods on texts with a minimal amount of noise. Since all three judges dealt with various topics, it is likely that successful methods would have to consider each author's style, rather than rely solely on content features in the texts.

Table 3.4 shows some dataset statistics of the Judgement dataset in comparison to the PAN'11 and Blog datasets (which are discussed in subsequent sections). As we can see, the Judgement dataset contains less authors than PAN'11 and Blog, but these authors wrote more texts than the average author in the two other datasets. Judgements are also substantially longer than the texts in all the other datasets (IMDb62, IMDb1M, PAN'11 and Blog), which should make authorship attribution on the Judgement dataset relatively easy compared to the other datasets we considered.

3.3.4 PAN'11

The PAN'11 datasets were introduced as part of the PAN 2011 competition (available from `pan.webis.de`) (Argamon and Juola, 2011). These datasets were extracted from the Enron email corpus (`www.cs.cmu.edu/~enron`), and were designed to emulate closed-class and open-class authorship attribution and authorship verification scenarios (Section 2.3.1). These datasets represent authorship attribution scenarios that may arise in computer forensics, such as the case noted by Chaski (2005), where an employee who was terminated for sending a racist email claimed that any person with access to his computer could have sent the email.

We used the largest PAN'11 dataset, with emails by 72 authors. Unlike the other datasets we used, this dataset is split into *training*, *validation* and *testing* subsets. We focused on the closed-class problem, using the validation and testing sets that contain texts only by training authors. The only change we made to the original dataset was dropping two training and two validation texts that were automatically

[7] We removed numbers and quotes by matching regular expressions for numbers and text in quotation marks, respectively.

	Judgement	**PAN'11**	**Blog**
Authors	3	72	19,320
Texts	1,342	Training: 9,335 Validation: 1,296 Testing: 1,300	678,161
Texts per author mean (stddev)	Dixon: 902 McTiernan: 253 Rich: 187	Training: 129.7 (139.3) Validation: 19.9 (19.0) Testing: 20.3 (18.9)	35.1 (105.0)
Tokens per text mean (stddev)	Dixon: 2,858.6 (2,456.9) McTiernan: 1,310.7 (1,248.4) Rich: 783.0 (878.5)	Training: 60.8 (109.4) Validation: 65.3 (98.9) Testing: 71.0 (115.1)	248.4 (510.8)

Table 3.4: Statistics for authorship-only datasets

generated, which were detected by length and content. This had a negligible effect on method accuracy, but made the statistics in Table 3.4 more representative of the data (e.g., the mean count of tokens per text is 65.3 in the validation set without the two automatically-generated texts, compared to 338.3 in the full validation set).

Using this dataset allows us to test our methods on short and informal texts with more authors than in traditional authorship attribution. As Table 3.4 shows, the PAN'11 dataset contains the shortest texts of the datasets we considered. This fact, together with the training/validation/testing structure of the dataset, make it possible to run many experiments on this dataset before moving on to larger datasets, such as the Blog dataset.

3.3.5 Blog

The Blog dataset is the largest dataset we considered, containing 678,161 blog posts by 19,320 authors (available from `u.cs.biu.ac.il/~koppel`). It was created by Schler et al. (2006) to learn about the relation between language use and demographic characteristics, such as age and gender. We use this dataset to test how our authorship attribution methods scale to handle thousands of authors. As blog posts can be about any topic, this dataset is less restricted than the IMDb, Judgement and PAN'11 datasets. Further, the large number of authors ensures that every topic is likely to interest at least several authors, meaning that methods that rely only on content are unlikely to perform as well as methods that also take author style into account.

3.3.6 MovieLens

The MovieLens datasets are commonly used to evaluate rating prediction algorithms (available from `www.grouplens.org`). Three MovieLens datasets were released. Their names correspond to the number of ratings each dataset contains: 100K,

Property	Value
Users	6,040
Items	3,706
User-item rating matrix sparsity	95.53%
Ratings	1,000,209
Ratings per user mean (stddev)	165.6 (192.8)
Ratings per item mean (stddev)	269.9 (384.1)

Table 3.5: MovieLens dataset statistics

1M and 10M. Only the smaller two contain demographic information about the users, and thus we chose to use the 1M dataset (we refer to MovieLens1M as the MovieLens dataset throughout this thesis). This demographic information includes each user's age, gender, occupation and postcode. It is worth noting that no texts by the users are included, meaning that we can use this dataset to test how our rating prediction approach performs when explicit user attributes are available (Chapter 7), but it cannot be used for testing the authorship attribution and polarity inference methods.

Table 3.5 presents some dataset statistics. When comparing the MovieLens statistics to the IMDb1M statistics (Table 3.3), it appears that rating prediction on MovieLens would be easier than on IMDb1M. This is because the MovieLens user-item rating matrix is not as sparse as the IMDb1M matrix, and it contains less users and items, but many more ratings per user and per item. It is worth noting that although only users with at least 20 ratings were included in the MovieLens dataset, we employ the GivenX protocol (Section 3.1) to emulate scenarios where the number of ratings per target user is low.

3.4 Preprocessing and External Tools

We applied minimal preprocessing to the texts in the datasets, and converted them into tokens using the default English sentence detector and tokeniser from OpenNLP 1.4.3 (`opennlp.sourceforge.net`). The preprocessing steps included:

- Replacing non-standard punctuation marks with standard ASCII punctuation, to make them recognisable by OpenNLP (e.g., curved quotation marks such as “ were converted to ").
- Breaking URLs into their components to make it easier to find commonalities across texts (e.g., “www.youtube.com/abc” was converted to “www youtube com abc”).
- Converting the texts to lower-case to reduce the number of features. This was performed after the sentence detection phase.

In our experiments, we used our own Java implementations of the topic models (Chapter 4), authorship attribution methods (Chapter 5), sentiment analysis framework (Chapter 6), and rating prediction algorithms (Chapter 7). In addition, we used the support vector machine (SVM) and support vector regression (SVR) implementations from Weka 3.6.0 (`www.cs.waikato.ac.nz/ml/weka`), with the linear kernel. As Weka's SVM implementation employs the one-versus-one (OVO) setup for multi-class problems, we also used L2-regularised linear SVMs from LIBLINEAR 1.8 (Fan et al., 2008), which uses the one-versus-all (OVA) setup, and is well-suited for large-scale text classification. For both SVM and SVR, we scaled the feature values to the $[0, 1]$ range, as advocated by Hsu et al. (2003) (we verified that this step improves performance in preliminary experiments). We experimented with cost parameter values from the set $\{\ldots, 10^{-1}, 10^{0}, 10^{1}, \ldots\}$, until no improvement in performance was obtained (starting from $10^{0} = 1$). In each experiment, we report the results obtained with the cost value that yielded the best performance, which gives an optimistic estimate for the performance of SVM baselines.

Chapter 4

Topical User Models

In recent years, topic models have gained popularity as a means of analysing large text corpora (Section 2.2). In this chapter, we suggest ways of using topic models to obtain compact representations of users' interests and characteristics. We first examine two popular topic models – *Latent Dirichlet Allocation* (LDA) and the *Author-Topic* (AT) model – in light of our user modelling goals. Then, we introduce the *Disjoint Author-Document Topic* (DADT) model – a topic model that we developed, which draws on the strengths of LDA and AT, while addressing their limitations by integrating them into a single model.

As discussed in Chapter 1, using topic models for user modelling is one of the key contributions of this thesis. As the name suggests, topic models are traditionally used to discover topics in text corpora. However, these "topics" are merely distributions over words, which do not necessarily correspond to a human-interpretable notion of topics. The meaning of the topics largely depends on the type of words that appear in the corpus. Specifically, in many studies stopwords and punctuation are removed from the corpus in a preprocessing step, allowing the inferred topics to be easily interpreted by humans by examining a list of the most probable words for each topic (Section 2.2). By contrast, we do not filter out any tokens in most of our experiments, since we are interested in models that capture both user interests and their authorship style (as mentioned in Section 2.3, stopwords are associated with authorship style, which is an indicator of user characteristics such as demographics and personality). Hence, our approach yields topics that are sometimes hard to interpret, but are nonetheless useful.

This chapter offers only a limited comparison of the topic models we considered, since we believe that – when possible – performance should be evaluated in the context of the actual task for which the models are used (Section 2.2.4). Ideally, we want a scalable model that yields user representations that are suitable when the goal is to discriminate between users according to texts they wrote, as in the authorship attribution task (Chapter 5), but also generates a soft clustering of users based

on their texts, as required for our polarity inference and rating prediction frameworks (Chapters 6 and 7 respectively).[1] Hence, we offer a more rigorous comparison of the models in subsequent chapters, in light of the task at hand. Specifically, we use the models presented in this chapter for the following purposes:

- Chapter 5: Document dimensionality reduction, measuring author and document distance, and inferring author probability for authorship attribution. We see authorship attribution as the primary testbed for using the topic models as user models, since authorship style is indicative of user characteristics (Section 2.3).
- Chapter 6: Measuring user similarity for polarity inference.
- Chapter 7: Obtaining compact text-based user representations for rating prediction.

This chapter is organised as follows. Section 4.1 introduces notation and provides a preliminary discussion of the meaning of the parameters used by the topic models. Section 4.2 delves into the technical details of LDA and AT, and describes our approach to using these models for user modelling. Section 4.3 introduces DADT, and provides a theoretical comparison of DADT to LDA and AT. Section 4.4 presents the results of our empirical comparison of LDA, AT and DADT on a synthetic dataset. Section 4.5 discusses two possible approaches to considering word order in the context of topical user modelling, and Section 4.6 concludes the chapter.

4.1 Preliminaries and Notation

This section introduces parameters whose values are given as input to the models discussed in this chapter (LDA, AT and DADT). The values of these parameters are either determined by the corpus (Section 4.1.1) or configured when using the models (Section 4.1.2). In addition to defining the parameters, we discuss practical considerations for setting the values of the configurable parameters.

Throughout this thesis, we denote matrices with uppercase boldface italics (e.g., $\boldsymbol{M}$), vectors with lowercase boldface italics (e.g., $\boldsymbol{v}$), and vector elements with lowercase italics with subscript index (e.g., v_i). The element at the i-th row and j-th column of $\boldsymbol{M}$ is denoted m_{ij}, and sets are denoted with calligraphic font (e.g., $\mathcal{S}$).

4.1.1 Corpus-dependent Parameters

The following parameters depend on the corpus, and are thus considered to be observed:

- Scalars:

  A: Number of authors. We use $a \in \{1, \ldots, A\}$ to denote an author identifier.

[1] Soft clustering allows users to belong to multiple clusters, in contrast to hard clustering where each user belongs to a single cluster.

D: Number of documents. We use $d \in \{1, \ldots, D\}$ to denote a document identifier.

V: Vocabulary size. We use $v \in \{1, \ldots, V\}$ to denote a unique word identifier in the vocabulary.

N_d: Number of words in document d. We use $i \in \{1, \ldots, N_d\}$ to denote a word index in document d.

- Stacked vectors:

$\boldsymbol{A}$: Document authors – a D-dimensional vector of vectors, where the d-th element $\boldsymbol{a}_d$ contains the authors of the d-th document. In cases where the corpus contains only single-authored texts, we use the scalar a_d to denote the author of the d-th document, since $\boldsymbol{a}_d$ is always of unit length.

$\boldsymbol{W}$: Document words – a D-dimensional vector of vectors, where the d-th element $\boldsymbol{w}_d$ contains the words of the d-th document. The vector $\boldsymbol{w}_d$ is of length N_d, and $w_{di} \in \{1, \ldots, V\}$ is the i-th word in the d-th document.

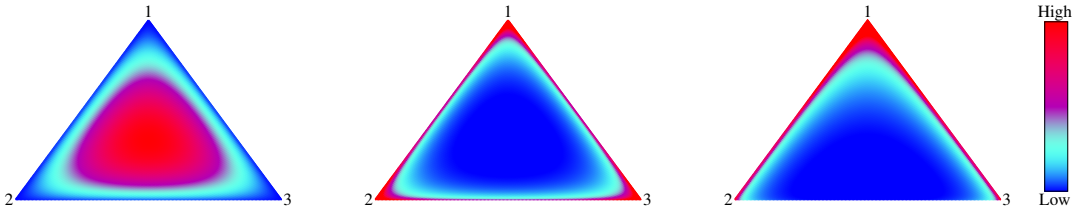
4.1.2 Configurable Parameters

Number of Topics

We make a distinction between *document* topics and *author* topics. In both cases the word "topic" describes a distribution over all the words in the vocabulary. The difference is that document topics are word distributions that arise from documents, while author topics are word distributions that characterise the authors. LDA uses only document topics, while AT uses only author topics (Section 4.2). DADT, our hybrid model, employs both document topics and author topics (Section 4.3).

All three models take the number of topics as a configurable parameter, denoted by $T^{(D)}$ for the number of document topics and by $T^{(A)}$ for the number of author topics. While the models have other configurable parameters (introduced below), we found that the number of topics has the largest impact on model performance because it controls the overall model complexity. For example, setting $T^{(D)} = 1$ in LDA means that all the words in all the documents are drawn from the same topic (i.e., distribution over words), while setting $T^{(D)} = 200$ gives LDA much more freedom to adapt to the corpus, as each word can be drawn from one of 200 topics.

It is worth noting that techniques for determining the optimal number of topics have been suggested. For example, Teh et al. (2006) used hierarchical Dirichlet processes to learn the number of topics while inferring the LDA model. We did not experiment with such techniques as they tend to complicate model inference, and we found that using a constant number of topics yields good performance. Nonetheless, we note that employing such techniques may be a worthwhile future



(a) $(2, 2, 2)$ (b) $(0.5, 0.5, 0.5)$ (c) $(0.9, 0.5, 0.5)$

Figure 4.1: Three-dimensional Dirichlet probability density, given three prior vectors

research direction, especially to determine the balance between document topics and author topics for our DADT model (Section 4.3).

Distribution Priors

The following parameters are the priors of the Dirichlet and beta distributions used by the models. In contrast to the number of topics, which controls model complexity, the priors allow users of the models to specify any prior knowledge and beliefs they may have about the data. Unlike the number of topics, which imposes a rigid constraint on the model, the effect of the priors on the inferred model is expected to diminish as the amount of observed data is increased (Equations 4.3, 4.4 and 4.5). Indeed, we found in our experiments that varying prior values had a small effect on performance compared to varying the number of topics.

The priors are defined as follows (all vector elements and scalars are positive):

- Priors used by LDA and DADT:

  $\boldsymbol{\alpha}^{(D)}$: Document topic prior – a vector of length $T^{(D)}$.
  $\boldsymbol{\beta}^{(D)}$: Prior for words in document topics – a vector of length V.

- Priors used by AT and DADT:

  $\boldsymbol{\alpha}^{(A)}$: Author topic prior – a vector of length $T^{(A)}$.
  $\boldsymbol{\beta}^{(A)}$: Prior for words in author topics – a vector of length V.

- Priors used only by DADT:

  $\delta^{(D)}$: Document words in document prior.
  $\delta^{(A)}$: Author words in document prior.
  $\boldsymbol{\eta}$: Author in corpus prior – a vector of length A.

The support of a K-dimensional Dirichlet distribution is the set of K-dimensional vectors with elements in the range $(0, 1)$ whose sum is 1 (in the two-dimensional case, it is equivalent to the beta distribution). Hence, each draw from the Dirichlet distribution can be seen as defining the parameters of a categorical distribution. This is illustrated by Figure 4.1, which shows the Dirichlet distribution density in the three-dimensional case for three different prior vectors (the density is triangular

because the drawn vector elements have to sum to 1 – each corner of the triangle corresponds to a dimension of the distribution). When the prior vector is symmetric, i.e., all its elements have the same value, the density is also symmetric (Figures 4.1a and 4.1b). Symmetric priors with element values that are greater than 1 yield densities that are concentrated in the middle of the triangle, meaning that categorical vectors with relatively uniform values are likely to be drawn (Figure 4.1a). On the other hand, symmetric priors with element values that are less than 1 yield sparse densities with high values in the corners of the triangle, meaning that the categorical vectors are likely to have one element whose value is greater than the other elements (Figure 4.1b). Finally, when the prior is asymmetric, vectors that give higher probabilities to the elements with higher prior values are likely to be drawn (Figure 4.1c).

The document and author topic priors ($\boldsymbol{\alpha}^{(D)}$ and $\boldsymbol{\alpha}^{(A)}$ respectively) encode our beliefs about the document and author topic distributions respectively. They are often set to be symmetric, since we have no reason to favour one topic over the other before we have seen the data (Steyvers and Griffiths, 2007). Wallach et al. (2009a) argued that employing asymmetric priors in LDA is beneficial, and suggested a method that learns such priors as part of model inference (by placing another prior on the $\boldsymbol{\alpha}^{(D)}$ prior). We implemented Wallach et al.'s method for all the models we considered, but found that it did not improve authorship attribution accuracy in preliminary experiments. Thus, in all our experiments we set the elements of $\boldsymbol{\alpha}^{(D)}$ and $\boldsymbol{\alpha}^{(A)}$ to $\min(0.1, 5/T^{(D)})$ and $\min(0.1, 5/T^{(A)})$ respectively, yielding relatively sparse topic distributions, since we expect each document and author to be sufficiently represented by only a few topics.[2]

The priors for words in document and author topics ($\boldsymbol{\beta}^{(D)}$ and $\boldsymbol{\beta}^{(A)}$ respectively) encode our beliefs about the word distributions. As for the topic distribution priors, symmetric priors are often used, with a default value of 0.01 for all the vector elements (yielding sparse word distributions, as indicated above), meaning that each topic is expected to assign high probabilities to only a few top words (Steyvers and Griffiths, 2007). In contrast to the topic distribution priors, Wallach et al. (2009a) found in their experiments on LDA that using an asymmetric $\boldsymbol{\beta}^{(D)}$ was of no benefit. This is because using an asymmetric $\boldsymbol{\beta}^{(D)}$ means that we encode a prior preference for a certain word to appear in all topics (e.g., a word represented by corner 1 in Figure 4.1c). For the same reason, using a symmetric $\boldsymbol{\beta}^{(A)}$ is a sensible choice for AT. In contrast to LDA and AT, our DADT model distinguishes between document words and author words, and thus employs both $\boldsymbol{\beta}^{(D)}$ and $\boldsymbol{\beta}^{(A)}$ as priors. This allows us to encode our prior knowledge that stopword use is indicative of authorship. Thus,

[2]We chose this value following the recommendations from LingPipe's documentation (`alias-i.com/lingpipe`), which are based on empirical evidence from several corpora.

for DADT we set $\beta_v^{(D)} = 0.01 - \epsilon$ and $\beta_v^{(A)} = 0.01 + \epsilon$ for all v, where v is a stopword (ϵ can be set to zero to obtain symmetric priors).[3]

DADT's $\delta^{(D)}$ and $\delta^{(A)}$ priors encode our prior belief about the balance between document words and author words in a given document. Document words (drawn from document topics) are expected to be representative of the documents in the corpus, while author words (drawn from author topics) characterise the authors in the corpus.[4] According to DADT's definition (Section 4.3.1), which employs the beta distribution, the prior expected value of the portion of each document that is composed of author words is

$$\frac{\delta^{(A)}}{\delta^{(A)} + \delta^{(D)}} \tag{4.1}$$

with a variance of

$$\frac{\delta^{(A)}\delta^{(D)}}{\left(\delta^{(A)} + \delta^{(D)}\right)^2 \left(\delta^{(A)} + \delta^{(D)} + 1\right)} \tag{4.2}$$

In our experiments, we chose values for $\delta^{(D)}$ and $\delta^{(A)}$ by deciding on the expected value and variance, and solving the above equations for $\delta^{(D)}$ and $\delta^{(A)}$ (Section 4.4.3).

Finally, DADT's $\boldsymbol{\eta}$ prior determines the prior belief about an author having written a document (without looking at the actual words in the document). This prior is only used on documents with unobserved authors (Section 4.3.1). Hence, we use it only for authorship attribution (Section 5.2.5), since we assume that all the authors are observed in the other scenarios considered in this thesis. Since we have no reason to favour one author over the other, we use a symmetric prior, setting $\eta_a = 1$ for each author a, which yields a uniform probability density.

4.2 LDA and AT as Topical User Models

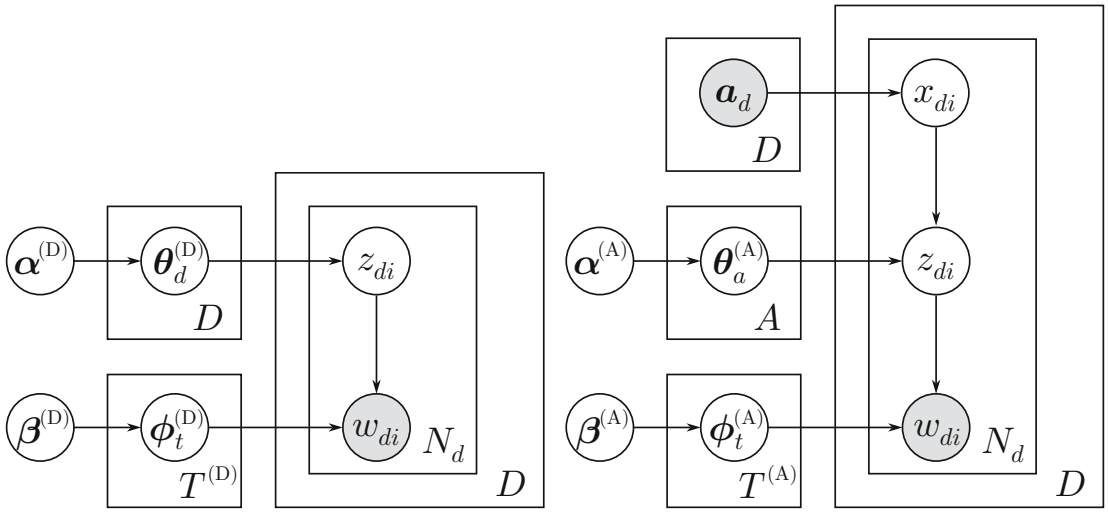
In this section, we suggest ways of using two existing topic models – LDA and AT – for user modelling. We go into the technical details of these models, expanding on the general descriptions from Section 2.2, as these two models form the basis of our DADT model, which we present in Section 4.3.

4.2.1 Model Definitions

Figure 4.2 presents LDA and AT in plate notation, where observed variables are in shaded circles, unobserved variables are in unshaded circles, and each box represents repeated sampling, with the number of repetitions at the bottom-right corner.

[3] The stopword list is provided in Appendix A.

[4] For example, if we asked two different authors to write a report about LDA, both reports are likely to contain content words like *Dirichlet*, *topic* and *prior*, but the frequencies of non-content words (e.g., function words and other indicators of authorship style) are likely to vary across the reports. In this case these content words are expected to be allocated to document topics, and the non-content words whose usage varies across authors would be allocated to author topics. In cases where the authors write about different issues, DADT may allocate some content words to author topics, i.e., the meaning of DADT's topics is expected to be corpus-specific.



(a) LDA (b) AT

Figure 4.2: Latent Dirichlet Allocation (LDA) and the Author-Topic (AT) model

LDA was originally defined by Blei et al. (2003). Here we describe Griffiths and Steyvers's (2004) extended version. The main idea behind LDA is that each document in a corpus is described by a distribution over topics, and each word in the document is drawn from its topic's word distribution. Formally, the generative process is as follows (Figure 4.2a):

- *Corpus level.*
  - For each topic $t \in \{1, \ldots, T^{(D)}\}$:
    - Draw a distribution over words $\boldsymbol{\phi}_t^{(D)} \sim \text{Dirichlet}\left(\boldsymbol{\beta}^{(D)}\right)$.
- *Document level.* For each document $d \in \{1, \ldots, D\}$:
  - Draw a distribution over topics $\boldsymbol{\theta}_d^{(D)} \sim \text{Dirichlet}\left(\boldsymbol{\alpha}^{(D)}\right)$.
  - *Word level.* For each word index $i \in \{1, \ldots, N_d\}$:
    - Draw the word's topic $z_{di} \sim \text{Categorical}\left(\boldsymbol{\theta}_d^{(D)}\right)$.
    - Draw the word from its topic's word distribution $w_{di} \sim \text{Categorical}\left(\boldsymbol{\phi}_{z_{di}}^{(D)}\right)$.

AT was introduced by Rosen-Zvi et al. (2004) to model author interests in corpora of multi-authored texts (e.g., research papers). The main idea behind AT is that each document is generated from the topic distributions of its observed authors, rather than from a document-specific topic distribution. Formally, the generative process is as follows (Figure 4.2b):

- *Corpus level.*
  - For each topic $t \in \{1, \ldots, T^{(A)}\}$:
    - Draw a distribution over words $\boldsymbol{\phi}_t^{(A)} \sim \text{Dirichlet}\left(\boldsymbol{\beta}^{(A)}\right)$.

  - For each author $a \in \{1, \ldots, A\}$:
    * Draw a distribution over topics $\boldsymbol{\theta}_a^{(A)} \sim \text{Dirichlet}\left(\boldsymbol{\alpha}^{(A)}\right)$.

- *Document level.* For each document $d \in \{1, \ldots, D\}$:
  - The document's set of authors $\boldsymbol{a}_d$ is observed.
  - *Word level.* For each word index $i \in \{1, \ldots, N_d\}$:
    * Draw the word's author x_{di} uniformly from $\boldsymbol{a}_d$.
    * Draw the word's topic $z_{di} \sim \text{Categorical}\left(\boldsymbol{\theta}_{x_{di}}^{(A)}\right)$.
    * Draw the word from its topic's word distribution $w_{di} \sim \text{Categorical}\left(\boldsymbol{\phi}_{z_{di}}^{(A)}\right)$.

A disadvantage of AT is that all the documents by the same authors are generated in an identical manner. To address this limitation, Rosen-Zvi et al. (2010) introduced "fictitious" authors, adding a unique "author" to each document. This allows AT to adapt itself to each document without changing the model specification.

4.2.2 Model Inference

Topic models are commonly inferred using either collapsed Gibbs sampling (Griffiths and Steyvers, 2004; Rosen-Zvi et al., 2004) or methods based on variational inference (Blei et al., 2003). We chose to employ collapsed Gibbs sampling to infer all models due to its efficiency and ease of implementation (Section 2.2.1). This involves repeatedly sampling from the conditional distribution of the latent parameters, which is obtained analytically by marginalising over the topic and word distributions, and using the properties of conjugate priors. These conditional distributions for LDA and AT are given in Equation 4.3 and Equation 4.4 respectively.[5]

$$p\left(z_{di} = t | \boldsymbol{W}, \boldsymbol{Z}_{-di}, \boldsymbol{\alpha}^{(D)}, \boldsymbol{\beta}^{(D)}\right) \propto \frac{\alpha_t^{(D)} + c_{dt}^{(DT)}}{\sum_{t'=1}^{T^{(D)}} \left(\alpha_{t'}^{(D)} + c_{dt'}^{(DT)}\right)} \frac{\beta_{w_{di}}^{(D)} + c_{tw_{di}}^{(DTV)}}{\sum_{v=1}^{V} \left(\beta_v^{(D)} + c_{tv}^{(DTV)}\right)} \tag{4.3}$$

$$p\left(x_{di} = a, z_{di} = t | \boldsymbol{A}, \boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Z}_{-di}, \boldsymbol{\alpha}^{(A)}, \boldsymbol{\beta}^{(A)}\right) \propto \frac{\alpha_t^{(A)} + c_{at}^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + c_{at'}^{(AT)}\right)} \frac{\beta_{w_{di}}^{(A)} + c_{tw_{di}}^{(ATV)}}{\sum_{v=1}^{V} \left(\beta_v^{(A)} + c_{tv}^{(ATV)}\right)} \tag{4.4}$$

where $\boldsymbol{Z}_{-di}$ and $\boldsymbol{X}_{-di}$ are all the topic and author assignments respectively, excluding the assignment for the i-th word of the d-th document. In addition, $c_{tw_{di}}^{(DTV)}$ and $c_{tw_{di}}^{(ATV)}$

[5] Griffiths and Steyvers (2004) and Steyvers and Griffiths (2007) provide more details on the derivation of Equation 4.3, and Rosen-Zvi et al. (2004, 2010) discuss the derivation of Equation 4.4. According to Rosen-Zvi et al. (2010), joint sampling of x_{di} and z_{di} yields faster convergence than separate sampling.

Parameter	Posterior Distribution	Expected Value
$\boldsymbol{\theta}_d^{(D)}$	Dirichlet $\left(\boldsymbol{\alpha}^{(D)} + \boldsymbol{c}_d^{(DT)}\right)$	$\mathbb{E}[\theta_{dt}^{(D)}] = \frac{\alpha_t^{(D)} + c_{dt}^{(DT)}}{\sum_{t'=1}^{T^{(D)}} \left(\alpha_{t'}^{(D)} + c_{dt'}^{(DT)}\right)}$
$\boldsymbol{\phi}_t^{(D)}$	Dirichlet $\left(\boldsymbol{\beta}^{(D)} + \boldsymbol{c}_t^{(DTV)}\right)$	$\mathbb{E}[\phi_{tv}^{(D)}] = \frac{\beta_v^{(D)} + c_{tv}^{(DTV)}}{\sum_{v'=1}^{V} \left(\beta_{v'}^{(D)} + c_{tv'}^{(DTV)}\right)}$
$\boldsymbol{\theta}_a^{(A)}$	Dirichlet $\left(\boldsymbol{\alpha}^{(A)} + \boldsymbol{c}_a^{(AT)}\right)$	$\mathbb{E}[\theta_{at}^{(A)}] = \frac{\alpha_t^{(A)} + c_{at}^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + c_{at'}^{(AT)}\right)}$
$\boldsymbol{\phi}_t^{(A)}$	Dirichlet $\left(\boldsymbol{\beta}^{(A)} + \boldsymbol{c}_t^{(ATV)}\right)$	$\mathbb{E}[\phi_{tv}^{(A)}] = \frac{\beta_v^{(A)} + c_{tv}^{(ATV)}}{\sum_{v'=1}^{V} \left(\beta_{v'}^{(A)} + c_{tv'}^{(ATV)}\right)}$

Table 4.1: LDA and AT expected values for the topic and word distributions

are the counts of word w_{di} in document or author topic t respectively, $c_{dt}^{(DT)}$ is the count of topic t in document d, and $c_{at}^{(AT)}$ is the count of topic t assignments to author a. Here, all the counts exclude the di-th assignments (i.e., x_{di} and z_{di}).

Commonly, several Gibbs sampling chains are run, and several samples are retained from each chain after a burn-in period, which allows the chain to reach its stationary distribution. For each sample, the topic distributions and the word distributions are estimated using their expected values, given the topic assignments $\boldsymbol{Z}$ and the author assignments $\boldsymbol{X}$. These expected values are given in Table 4.1 (here, the counts are over the full topic and author assignments). All the posterior distributions take a similar form due to the fact that the Dirichlet distribution is the conjugate prior of the categorical distribution (Griffiths and Steyvers, 2004; Rosen-Zvi et al., 2004). Note that these values cannot be averaged across samples due to the exchangeability of the topics (Steyvers and Griffiths, 2007), e.g., topic 1 in one sample is not necessarily the same as topic 1 in another sample.

4.2.3 Application to User Modelling

LDA does not directly model the document authors (i.e., the users). Nevertheless, it can still be used to obtain valuable information about them. The output of LDA consists of distributions over topics $\boldsymbol{\theta}_d^{(D)}$ for each document d. As the number of topics $T^{(D)}$ is commonly much smaller than the size of the vocabulary V, these topical representations form a lower-dimensional representation of the corpus. The two LDA-based user models we consider in this thesis are (assuming all the texts were written by a single user):

- **LDA-M** (LDA with multiple user documents): This model represents each user as the set of distributions over topics of their documents, i.e., for a user u it is the set $\left\{\boldsymbol{\theta}_d^{(D)} | a_d = u\right\}$.
- **LDA-S** (LDA with a single user document): This model adds a preprocessing step where each user's documents are concatenated into a single document.

Then, LDA is run on the concatenated documents. Each user is thus represented by a single distribution over topics (the distribution of the concatenated document).[6]

An advantage of LDA-S over LDA-M is that LDA-S yields a much more compact user representation than LDA-M, especially for users who authored many documents. However, this compactness may come at the price of accuracy, as markers that may be present only in a few short documents by one user may lose their prominence if these documents are concatenated with longer documents.

AT naturally yields user models in the form of distributions over topics. That is, each user u is represented as a distribution over topics $\boldsymbol{\theta}_u^{(A)}$. Since AT can be run with fictitious authors, we consider the two following variants:

- **AT**: "Pure" AT, without fictitious authors.
- **AT-FA**: AT, when run with the additional preprocessing step of adding a fictitious author to each document, which is meant to allow the model to adapt to different documents (Section 4.2.1).

It is worth noting that when analysing single-authored documents, AT is equivalent to LDA-S. Our main focus in this thesis is on single-authored texts. However, we presented both LDA-S and AT because practitioners may find it easier to employ LDA-S due to the relative prevalence of LDA implementations.[7] Nonetheless, throughout this thesis we only present results obtained with LDA-M, AT, and AT-FA, since we use our own Java implementations of these models.

4.3 DADT: A Hybrid Model

Our DADT model can be seen as a combination of LDA and AT, which is meant to address the weaknesses of both models while retaining their strengths. The main idea behind DADT is that words are generated from two disjoint sets of topics: document topics and author topics. Words generated from document topics follow the same generation process as in LDA, while words generated from author topics are generated in an AT-like fashion. This approach has the potential benefit of separating "document" words from "author" words. That is, words whose use varies across documents are expected to be found in document topics, while words whose use varies between authors are expected to be assigned to author topics.

We present the formal model definition in Section 4.3.1, and develop the inference algorithm in Section 4.3.2. We then discuss the differences between DADT and LDA and AT in Section 4.3.3.

[6] Concatenating all the author documents into one document has been named the *profile-based* approach in previous authorship attribution studies, in contrast to the *instance-based* approach, where each document is considered separately (Stamatatos, 2009).

[7] In fact, our initial modelling approach was LDA-S for exactly this reason.

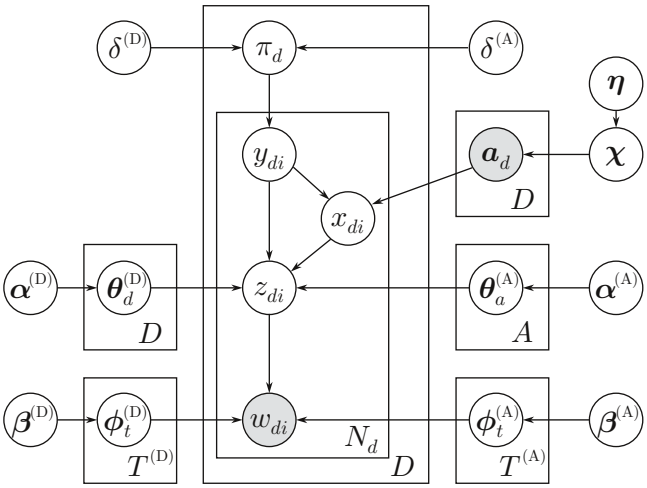


Figure 4.3: The Disjoint Author-Document Topic (DADT) model

4.3.1 Model Definition

Figure 4.3 presents the graphical representation of the model, where the document-dependent parameters appear on the left-hand side, and the author-dependent parameters appear on the right-hand side. Formally, the generative process is as follows (we mark each step as coming from either **L**DA or **A**T, or as new in **D**ADT).

- *Corpus level.*
  - **L.** For each document topic $t \in \{1, \ldots, T^{(D)}\}$:
    - Draw a distribution over words $\boldsymbol{\phi}_t^{(D)} \sim \text{Dirichlet}\left(\boldsymbol{\beta}^{(D)}\right)$.
  - **A.** For each author topic $t \in \{1, \ldots, T^{(A)}\}$:
    - Draw a distribution over words $\boldsymbol{\phi}_t^{(A)} \sim \text{Dirichlet}\left(\boldsymbol{\beta}^{(A)}\right)$.
  - **A.** For each author $a \in \{1, \ldots, A\}$:
    - Draw a distribution over topics $\boldsymbol{\theta}_a^{(A)} \sim \text{Dirichlet}\left(\boldsymbol{\alpha}^{(A)}\right)$.
  - **D.** Draw a distribution over authors $\boldsymbol{\chi} \sim \text{Dirichlet}\left(\boldsymbol{\eta}\right)$.
- *Document level.* For each document $d \in \{1, \ldots, D\}$:
  - **L.** Draw a distribution over document topics $\boldsymbol{\theta}_d^{(D)} \sim \text{Dirichlet}\left(\boldsymbol{\alpha}^{(D)}\right)$.
  - **D.** Draw the document's author set $\boldsymbol{a}_d$ by repeatedly sampling without replacement from Categorical $(\boldsymbol{\chi})$.[8]
  - **D.** Draw the document's author/document topic ratio $\pi_d \sim \text{Beta}\left(\delta^{(A)}, \delta^{(D)}\right)$.

[8]This can be modelled as sampling from Wallenius's noncentral hypergeometric distribution (Fog, 2008) with a weight vector $\boldsymbol{\chi}$ and a parameter vector whose elements are all equal to 1. In this thesis, we consider only situations where $\boldsymbol{a}_d$ is observed when the model is inferred. When handling documents with unknown authors in our authorship attribution experiments, we assume that all anonymous texts are single-authored (Section 5.1.4).

– *Word level.* For each word index $i \in \{1, \ldots, N_d\}$:

D. Draw the author/document topic word indicator $y_{di} \sim \text{Bernoulli}(\pi_d)$.

D. If $y_{di} = 0$, generate the word from the document topics:

L. Draw the word's topic $z_{di} \sim \text{Categorical}\left(\boldsymbol{\theta}_d^{(D)}\right)$.

L. Draw the word from its topic's word distribution $w_{di} \sim \text{Categorical}\left(\boldsymbol{\phi}_{z_{di}}^{(D)}\right)$.

D. If $y_{di} = 1$, generate the word from the author topics:

A. Draw the word's author x_{di} uniformly from $\boldsymbol{a}_d$.

A. Draw the word's topic $z_{di} \sim \text{Categorical}\left(\boldsymbol{\theta}_{x_{di}}^{(A)}\right)$.

A. Draw the word from its topic's word distribution $w_{di} \sim \text{Categorical}\left(\boldsymbol{\phi}_{z_{di}}^{(A)}\right)$.

4.3.2 Model Inference

We infer DADT using collapsed Gibbs sampling, as done for LDA and AT (Section 4.2.2). This involves repeatedly sampling from the following conditional distribution of the latent parameters, which is obtained analytically by marginalising over the topic and word distributions, and using the properties of conjugate priors (details of this derivation are given in Appendix B).

$$p\left(x_{di} = a, y_{di} = y, z_{di} = t \middle| \boldsymbol{A}, \boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \boldsymbol{\alpha}^{(D)}, \boldsymbol{\beta}^{(D)}, \delta^{(D)}, \boldsymbol{\alpha}^{(A)}, \boldsymbol{\beta}^{(A)}, \delta^{(A)}\right) \propto \tag{4.5}$$

$$\begin{cases} \left(\delta^{(D)} + c_d^{(DD)}\right) \dfrac{\alpha_t^{(D)} + c_{dt}^{(DT)}}{\sum_{t'=1}^{T^{(D)}} \left(\alpha_{t'}^{(D)} + c_{dt'}^{(DT)}\right)} \dfrac{\beta_{w_{di}}^{(D)} + c_{tw_{di}}^{(DTV)}}{\sum_{v=1}^{V} \left(\beta_v^{(D)} + c_{tv}^{(DTV)}\right)} & \text{if } y = 0 \\ \left(\delta^{(A)} + c_d^{(DA)}\right) \dfrac{\alpha_t^{(A)} + c_{at}^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + c_{at'}^{(AT)}\right)} \dfrac{\beta_{w_{di}}^{(A)} + c_{tw_{di}}^{(ATV)}}{\sum_{v=1}^{V} \left(\beta_v^{(A)} + c_{tv}^{(ATV)}\right)} & \text{if } y = 1 \end{cases}$$

where $\boldsymbol{Y}_{-di}$ contains the topic indicators, excluding the di-th value; and $c_d^{(DD)}$ and $c_d^{(DA)}$ are the counts of words assigned to document or author topics in document d, respectively. The other variables are defined as for LDA and AT (Section 4.2.2). Here, all the counts exclude the di-th assignments (i.e., x_{di}, y_{di} and z_{di}).

The building blocks of our DADT model are clearly visible in Equation 4.5. LDA's Equation 4.3 is contained in the first case, where the word is drawn from document topics ($y = 0$), while AT's Equation 4.4 is contained in the second case, where the word is drawn from author topics ($y = 1$). However, Equation 4.5 also demonstrates the main difference between DADT and its building blocks, as DADT

Parameter	Posterior Distribution	Expected Value
π_d	Beta $\left(\delta^{(A)} + c_d^{(DA)}, \delta^{(D)} + c_d^{(DD)}\right)$	$\mathbb{E}[\pi_d] = \frac{\delta^{(A)}+c_d^{(DA)}}{\delta^{(D)}+\delta^{(A)}+N_d}$
$\boldsymbol{\chi}$	Dirichlet $\left(\boldsymbol{\eta} + \boldsymbol{c}^{(AD)}\right)$	$\mathbb{E}(\chi_a) = \frac{\eta_a+c_a^{(AD)}}{\sum_{a'=1}^{A}\left(\eta_{a'}+c_{a'}^{(AD)}\right)}$

Table 4.2: DADT expected values for the author/document ratio and the corpus author distribution

considers *both* documents and authors during the inference process by assigning each word to either a document topic or an author topic, where document topics and author topics come from disjoint sets.

Algorithm 4.1 presents the full procedure for model inference. It assumes the existence of a random number generator that makes it possible to draw samples from uniform and categorical distributions. For brevity and to avoid introducing more notation, the algorithm recalculates the sums that appear in the denominators of Equation 4.5, but these sums can be cached in practice.

As for LDA and AT, we ran several sampling chains in our experiments, retaining samples from each chain after a burn-in period, which allows the chain to reach its stationary distribution (a sample consists of $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$). For each sample, the topic and word distributions are estimated using their expected values given the latent variable assignments. The expected values for the topic and word distributions are the same as for LDA and AT (Table 4.1), and the expected values for the author/document ratio and the corpus author distribution appear in Table 4.2 (here, the counts are over the full assignments $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$). As in LDA and AT, the posterior distributions were straightforward to obtain, since the Dirichlet distribution is the conjugate prior of the categorical distribution and the beta distribution is the conjugate prior of the Bernoulli distribution. It is worth noting that since we assume that the documents' authors are observed during model inference, the expected value of each element of the corpus distribution over authors χ_a does not vary across samples, as it only depends on the prior η_a and on author a's count of documents in the corpus $c_a^{(AD)}$.

4.3.3 Comparison to LDA and AT

DADT can be seen as a generalisation of LDA and AT – setting DADT's number of author topics $T^{(A)}$ to zero yields a model that is equivalent to LDA, and setting the number of document topics $T^{(D)}$ to zero yields a model that is equivalent to AT. An advantage of DADT over LDA and AT is that both documents and authors are accounted for in the model's definition. Hence, preprocessing steps such as concatenating each author's documents or adding fictitious authors – as done in

Algorithm 4.1 DADT Inference

Input: $\boldsymbol{W}, \boldsymbol{A}, T^{(D)}, T^{(A)}, \boldsymbol{\alpha}^{(D)}, \boldsymbol{\alpha}^{(A)}, \boldsymbol{\beta}^{(D)}, \boldsymbol{\beta}^{(A)}, \delta^{(D)}, \delta^{(A)}, \boldsymbol{\eta}$, *numSteps*
Output: $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$

Initialisation:
  Initialise $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{C}^{(DT)}, \boldsymbol{C}^{(DTV)}, \boldsymbol{C}^{(AT)}, \boldsymbol{C}^{(ATV)}, \boldsymbol{c}^{(DD)}$ and $\boldsymbol{c}^{(DA)}$ with zeroes
  for $d = 1$ **to** D **do**
    for $i = 1$ **to** N_d **do**
      $y_{di} \leftarrow \text{Uniform}(\{0, 1\})$
      if $y_{di} = 0$ **then**
        $z_{di} \leftarrow \text{Uniform}(\{1, \ldots, T^{(D)}\})$
      else
        $z_{di} \leftarrow \text{Uniform}(\{1, \ldots, T^{(A)}\})$
        $x_{di} \leftarrow \text{Uniform}(\boldsymbol{a}_d)$
      AdjustCounts$(d, i, +1)$

Main loop:
  Let $\boldsymbol{p}$ be a local vector of length $T^{(D)} + |\boldsymbol{a}_d| T^{(A)}$
  for *step* $= 1$ **to** *numSteps* **do**
    for $d = 1$ **to** D **do**
      for $i = 1$ **to** N_d **do**
        AdjustCounts$(d, i, -1)$
        for $t = 1$ **to** $T^{(D)}$ **do**
          $p_t \leftarrow p\left(x_{di} = 0, y_{di} = 0, z_{di} = t\right)$ (Equation 4.5)
        for $x = 1$ **to** $|\boldsymbol{a}_d|$ **do**
          for $t = 1$ **to** $T^{(A)}$ **do**
            $p_{T^{(D)}+(x-1)T^{(A)}+t} \leftarrow p\left(x_{di} = a_{dx}, y_{di} = 1, z_{di} = t\right)$ (Equation 4.5)
        $\Sigma_{\boldsymbol{p}} \leftarrow \sum_{j=1}^{|\boldsymbol{p}|} p_j$
        for $j = 1$ **to** $|\boldsymbol{p}|$ **do**
          $p_j \leftarrow p_j / \Sigma_{\boldsymbol{p}}$
        $j \leftarrow \text{Categorical}(\boldsymbol{p})$
        if $j \leq T^{(D)}$ **then**
          $y_{di} \leftarrow 0$
          $z_{di} \leftarrow j$
        else
          $y_{di} \leftarrow 1$
          $z_{di} \leftarrow 1 + \left(j - T^{(D)}\right) \bmod T^{(A)}$
          $x_{di} \leftarrow a_{d\lceil \left(j - T^{(D)}\right)/T^{(A)} \rceil}$
        AdjustCounts$(d, i, +1)$

AdjustCounts(d, i, c) *procedure:*
  if $y_{di} = 0$ **then**
    $c_d^{(DD)} \leftarrow c_d^{(DD)} + c$
    $c_{dz_{di}}^{(DT)} \leftarrow c_{dz_{di}}^{(DT)} + c$
    $c_{z_{di}w_{di}}^{(DTV)} \leftarrow c_{z_{di}w_{di}}^{(DTV)} + c$
  else
    $c_d^{(DA)} \leftarrow c_d^{(DA)} + c$
    $c_{x_{di}z_{di}}^{(AT)} \leftarrow c_{x_{di}z_{di}}^{(AT)} + c$
    $c_{z_{di}w_{di}}^{(ATV)} \leftarrow c_{z_{di}w_{di}}^{(ATV)} + c$

LDA (with a single author document) and AT-FA to obtain author and document representations respectively (Section 4.2.3) – are unnecessary.

Of the LDA and AT variants presented in Section 4.2.3, DADT might seem most similar to AT-FA. However, there are several key differences between DADT and AT-FA.

First, in DADT *author topics are disjoint from document topics*, with different priors for each topic set. Thus, the number of author topics $T^{(A)}$ can be different from the number of document topics $T^{(D)}$, enabling us to vary the number of author and document topics according to the number of authors and documents in the corpus. For example, in the Judgement dataset (Section 3.3.3), which includes only a few authors that wrote many long documents, we expect that small values of $T^{(A)}$ compared to $T^{(D)}$ would suffice to get good author representations. By contrast, modelling the 19,320 authors of the Blog dataset (Section 3.3.5) is expected to require many more author topics.[9]

Second, DADT *places different priors on the word distributions* for author topics and document topics ($\boldsymbol{\beta}^{(A)}$ and $\boldsymbol{\beta}^{(D)}$ respectively). We know from previous work that stopwords are strong indicators of authorship (Koppel et al., 2009). Our model allows us to encode this prior knowledge by giving elements that correspond to stopwords in $\boldsymbol{\beta}^{(A)}$ higher weights than such elements in $\boldsymbol{\beta}^{(D)}$ (Section 4.1.2). We demonstrate the merits of encoding such prior knowledge in our experiments on a synthetic dataset (Section 4.4). In addition, we found that this property of DADT has practical benefits, as it improved the accuracy of DADT-based authorship attribution methods in our experiments (Section 5.3).

Third, DADT *learns the ratio between document words and author words* on a per-document basis, and makes it possible to specify a prior belief of what this ratio should be. As for the previous point, we demonstrate this advantage of DADT on a synthetic dataset in Section 4.4. We also show that it has practical benefits in our authorship attribution experiments (Section 5.3): specifying a prior belief that on average, about 80% of each document is composed of author words can yield better results than using AT's fictitious author approach that evenly splits each document into author and document words.

Fourth, DADT *defines the process that generates authors*. This allows us to consider the number of texts by each author when performing authorship attribution (Chapter 5). In addition, this enables the potential use of DADT in a semi-supervised setup by training on documents with unknown authors – an extension that is left for future work (Section 8.2).

[9]It is worth noting that adding topics increases model complexity and thus adds to the runtime of the inference algorithm (Section 4.1.2). Hence, on large datasets using more than a few hundred topics may become too computationally expensive. Being able to specify the balance between document and author topics in such cases is beneficial (Section 5.3.4).

Document	Author	Word Count							
		bank	loan	money	of	river	stream	the	to
1	1	4	4	4	2	0	0	5	2
2	2	5	7	9	6	0	0	1	2
3	1	7	4	5	2	0	0	5	2
4	2	7	5	4	5	0	0	2	2
5	1	7	5	4	2	0	0	5	2
6	2	9	4	3	6	0	0	1	2
7	1	4	5	6	2	1	0	5	2
8	2	6	3	4	5	1	2	2	2
9	1	6	2	4	2	1	3	5	2
10	2	6	4	1	5	2	3	2	2
11	1	7	1	3	2	2	3	5	2
12	2	6	0	1	5	3	5	2	2
13	1	6	1	0	2	6	3	5	2
14	2	6	0	0	5	2	5	2	2
15	1	5	0	0	2	4	10	5	2
16	2	4	0	0	5	5	7	2	2

Table 4.3: Our synthetic dataset

4.4 Model Comparison Using a Synthetic Dataset

In this section, we compare LDA, AT, and DADT empirically through experiments on a simple synthetic dataset, which makes the results relatively straightforward to analyse. The purpose of this section is to give a general feeling for the differences between the models and for the meaning of the inferred topics. While several approaches to evaluating the performance of topic models have been suggested (Section 2.2.4), we believe that – when possible – performance should be evaluated in the context of the actual task for which the models are used. Such a comparison of the models is offered in subsequent chapters.

4.4.1 The Dataset

The dataset we use here is based on the dataset Steyvers and Griffiths (2007) used to demonstrate the performance of LDA. We extended the original dataset by adding authors and function words. Our dataset, presented in Table 4.3, is designed to contain two document topics, one money-related and the other river-related, and two author topics, one characterising Author 1 and the other characterising Author 2. The documents are sorted by their main topic, with strongly money-related documents at the top and strongly river-related documents at the bottom. Documents by Author 1 are odd-numbered while Author 2's documents are even-numbered. Since

the models we consider here do not take word order into account, we only show each document's word counts. We use the following colour scheme to help identify the words in the dataset:

- The content word bank is coloured blue-green. It can be used either in the sense of a bank where one deposits money (money-related document topic), or a river bank (river-related document topic).
- The content words loan and money are coloured green, and are expected to be allocated to the money-related document topic.
- The function word of is coloured red. It is used by both authors, but Author 2 tends to use it more often than Author 1.
- The content words river and stream are coloured blue, and are expected to be allocated to the river-related document topic.
- The function word the is coloured pink. It is used by both authors, but Author 1 tends to use it more often than Author 2.
- The function word to is coloured orange, and is used by both authors with equal frequencies.

It is worth noting that in the general case, e.g., as in the Blog corpus (Section 3.3.5), authors may vary in their use of content words according to their interests. Our synthetic dataset represents a scenario where the authors write about the same content, but vary in authorship style, which is reflected in this case only by their use of function words.

4.4.2 Experimental Setup

For each model, we ran a single Gibbs chain for 1,000 iterations. We display the word assignments to topics from the last sample in the chain, and discuss the model characteristics that it demonstrates. While in general it is good practice to use several chains and take several samples from each chain, we found that the results are rather stable across chains and samples. In addition, since the topics are exchangeable, we cannot average topic counts across different samples (Section 4.2.2).

To ensure a fair comparison, all the models were run with the same number of overall topics. This number was set to four, since we expect to see two topics that characterise the documents and two topics that characterise the authors. Unless otherwise specified, we used symmetric priors of $\alpha_t^{(D)} = \alpha_t^{(A)} = 0.1$ for each topic t, and $\beta_v^{(D)} = \beta_v^{(A)} = 0.01$ for each word v (Section 4.1.2).

4.4.3 Results

Table 4.4 shows the log-likelihood of the dataset given the inferred model for each set of parameters. The log-likelihood can be seen as a general measure of fit, with

Model	Parameters					Log-likelihood
	$T^{(D)}$	$T^{(A)}$	$\delta^{(D)}$	$\delta^{(A)}$	ϵ	
LDA	4	—	—	—	—	-729.73
AT	—	4	—	—	—	-791.86
AT-FA	—	4	—	—	—	-727.38
DADT	2	2	1	1	0	-714.60
	2	2	4.889	1.222	0	-719.92
	2	2	4.889	1.222	0.009	-719.28

Table 4.4: Log-likelihoods of the synthetic dataset given each model's sample

Topic	Word Assignments								Total
	bank	loan	money	of	river	stream	the	to	
1	13	0	0	9	27	41	17	0	107
2	10	0	0	0	0	0	0	0	10
3	21	0	0	24	0	0	0	13	58
4	51	45	48	25	0	0	37	19	225

Table 4.5: LDA synthetic dataset sample topics

higher (i.e., less negative) values implying a better fit of the model to the data. However, models with high log-likelihood are not necessarily "better" in terms of interpretability or suitability for a certain task.

LDA

Table 4.5 shows the topics obtained by LDA. As expected, LDA discovered the river-related and money-related document topics (topics 1 and 4 respectively). More words were assigned to topic 4 since there are more purely money-related documents in the dataset than purely river-related documents (Table 4.3). The two other topics are harder to interpret. Topic 2 is relatively empty, with only 10 occurrences of the word "bank", which occurs in all the documents. Topic 3 contains many occurrences of the word "of", which Author 2 tends to use more often than Author 1, but there is no topic that captures Author 1's frequent use of the word "the". This relatively poor allocation of author words to topics was to be expected, since LDA does not take authors into account.

AT

Table 4.6 shows the topics obtained by AT. Since AT does not take the documents into account, content words were distributed between topics in a manner that is hard to interpret. Specifically, topic 1 contains all the occurrences of the words "bank", "loan", "stream", and "to" – it does not represent the themes we want

Topic	**Word Assignments**								**Total**
	bank	loan	money	of	river	stream	the	to	
1	95	45	17	0	0	41	0	32	230
2	0	0	31	0	0	0	54	0	85
3	0	0	0	58	0	0	0	0	58
4	0	0	0	0	27	0	0	0	27

Table 4.6: AT synthetic dataset sample topics

Topic	**Word Assignments**								**Total**
	bank	loan	money	of	river	stream	the	to	
1	1	0	0	0	27	41	0	0	69
2	0	0	0	0	0	0	54	31	85
3	0	45	48	0	0	0	0	0	93
4	94	0	0	58	0	0	0	1	153

Table 4.7: AT-FA synthetic dataset sample topics

to see. However, the words "the" and "of" are allocated to two different topics (2 and 3 respectively), yielding the author-related word allocation we expected. The poor document representation obtained by AT is reflected in it having the lowest log-likelihood of the models we considered (Table 4.4).

AT-FA

Table 4.7 presents the topics obtained by AT when run with fictitious authors. These topics form a better representation of the dataset than both LDA and AT. Topics 1 and 3 capture the content words, with topic 1 representing the river-related topic and topic 3 representing the money-related topic. Topics 2 and 4 are allocated the words that represent the authors, with topic 2 containing all occurrences of the word "the", and topic 4 containing all occurrences of the word "of". However, the words "bank" and "to" were not allocated in a satisfactory manner. We expected the content word "bank", which is used to describe both document topics, to be allocated to topics 1 and 3 rather than mostly to topic 4. Similarly, we expected the function word "to", which is used equally often by both authors, to be allocated more evenly to topics 2 and 4 rather than mostly to topic 2.

DADT

Table 4.8 presents the topics obtained by DADT, which we ran with the following parameter sets.

Topic	Word Assignments								Total
	bank	loan	money	of	river	stream	the	to	
D1	0	45	48	0	0	0	0	0	93
D2	0	0	0	0	27	41	0	0	68
A1	39	0	0	0	0	0	54	16	109
A2	56	0	0	58	0	0	0	16	130

(a) $\delta^{(D)} = \delta^{(A)} = 1$, $\epsilon = 0$

Topic	Word Assignments								Total
	bank	loan	money	of	river	stream	the	to	
D1	59	45	48	0	0	0	0	24	176
D2	36	0	0	0	27	41	0	8	112
A1	0	0	0	0	0	0	54	0	54
A2	0	0	0	58	0	0	0	0	58

(b) $\delta^{(D)} = 4.889$, $\delta^{(A)} = 1.222$, $\epsilon = 0$

Topic	Word Assignments								Total
	bank	loan	money	of	river	stream	the	to	
D1	56	45	48	0	0	0	0	0	149
D2	39	0	0	0	27	41	0	0	107
A1	0	0	0	0	0	0	54	18	72
A2	0	0	0	58	0	0	0	14	72

(c) $\delta^{(D)} = 4.889$, $\delta^{(A)} = 1.222$, $\epsilon = 0.009$

Table 4.8: DADT synthetic dataset sample topics

1. **Table 4.8a**. Uninformed priors on the document/author word ratio ($\delta^{(D)} = \delta^{(A)} = 1$) and symmetric word-in-topic priors ($\epsilon = 0$, as defined in Section 4.1.2). In this setup, DADT is similar to AT-FA, except for the document and author topic sets being disjoint.
2. **Table 4.8b**. Same as (1), with informed priors on the document/author word ratio that encode our prior knowledge that the synthetic dataset contains more content words than function words ($\delta^{(D)} = 4.889$ and $\delta^{(A)} = 1.222$ – obtained as described below).
3. **Table 4.8c**. Same as (2), with asymmetric word-in-topic priors that encode our prior knowledge that stopwords are more representative of author style than document content ($\epsilon = 0.009$).

As Table 4.8a shows, moving from AT-FA's approach of modelling all documents and authors over a single topic set to DADT's approach of using two disjoint topic

sets yielded a better representation of the dataset. Specifically, document topic 1 (D1 in the table) contains the money-related words and document topic 2 (D2) contains the river-related words. The function words were allocated to the author topics as expected, with author topic 1 (A1) containing all occurrences of the word "the" and half of the occurrences of the word "to", and author topic 2 (A2) allocated all occurrences of the word "of" and the remaining occurrences of the word "to". While this sample obtained the highest log-likelihood of those we experimented with (Table 4.4), it contains one flaw in terms of interpretability: the content word "bank" is allocated to the author topics rather than to document topics. This is probably because the word "bank" is used equally often for both content themes and by both authors, i.e., there is no strong evidence to suggest that it should be allocated to document topics. This can be addressed by tuning DADT's priors to encode our expectations. We describe the results of this tuning in the next two paragraphs.

In our second DADT experiment, we adjusted the document/author word ratio priors. In general, we do not know how many words in each document are document words and how many are author words. Even though we know the exact allocation of words to documents and authors in the synthetic dataset, for the second set of parameters we chose to specify priors that encode a general belief that about 80% of each document is composed of document words, while allowing for variability between documents with a standard deviation of 15% ($\delta^{(D)} = 4.889$ and $\delta^{(A)} = 1.222$, obtained by setting Equation 4.1 to 0.8 and Equation 4.2 to 0.15^2 and solving for $\delta^{(D)}$ and $\delta^{(A)}$, as described in Section 4.1.2). As Table 4.8b shows, specifying these priors had the desired effect of moving all occurrences of the word "bank" to document topics, but it also affected the word "to", which was also allocated solely to document topics. The fact that the other words were unaffected demonstrates the robustness of the model to small changes in the priors, i.e., words for which there is strong evidence in terms of occurrences in certain documents or use by certain authors were unaffected by our prior specification, while "bank" and "to", for which the evidence is weaker because they are distributed similarly between documents and authors, were affected by prior beliefs.

In our third DADT experiment, we encoded our prior knowledge about stopword use. As discussed in Section 4.1, DADT's use of two disjoint topic sets allows us to encode the prior belief that stopwords are more indicative of author style than document content by setting different priors for stopwords in document topics and in author topics. Table 4.8c shows the results of an experiment where we set $\epsilon = 0.009$ together with $\delta^{(D)} = 4.889$ and $\delta^{(A)} = 1.222$, meaning that for each function word v, $\beta_v^{(A)} = 0.019$ and $\beta_v^{(D)} = 0.001$ – encoding a prior belief that the function words should appear more frequently in author topics (Section 4.1.2). This

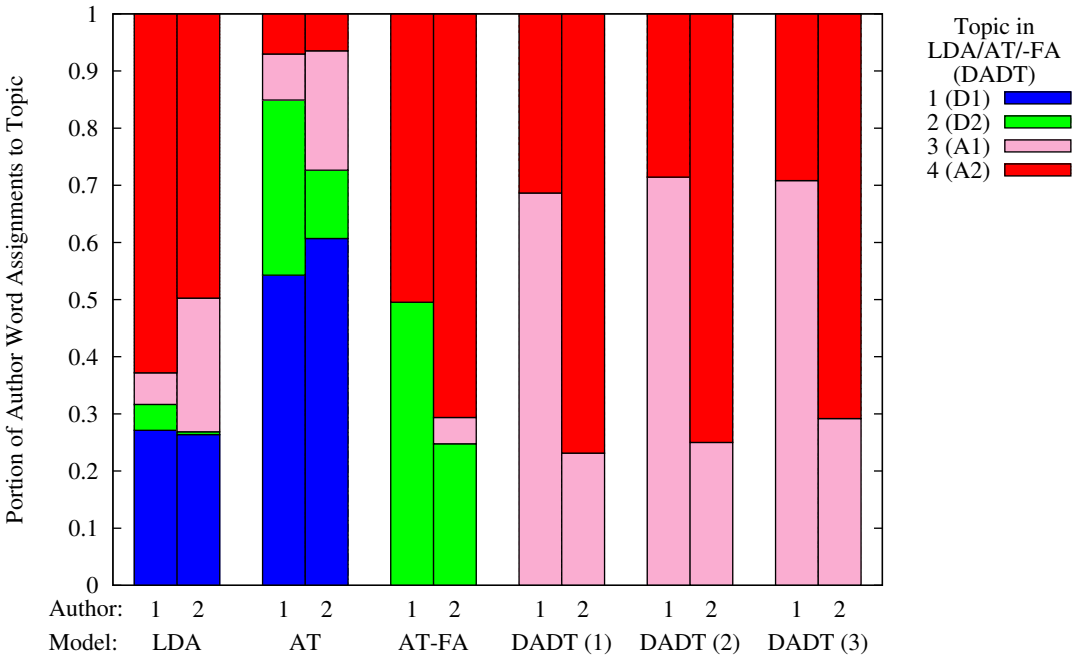


Figure 4.4: Comparison of author representations on the synthetic dataset

had the desired effect of allocating the word "to" to the author topics rather than to the document topics, as expected. As Table 4.4 shows, this had a minor effect on the log-likelihood, which indicates that from DADT's point of view, having the word "to" assigned to author topics or to document topics does not matter much in terms of fitting the observed data.

Comparison of Author Representations

Figure 4.4 compares the author representations yielded by the topic models. For each model, the two stacked bars represent the authors, and the four coloured sections represent the portion of the author words that were assigned to each topic (each topic is represented by a different colour). Since LDA does not consider authors, we simply assumed that all the words in the corpus are author words. The same assumption applies to AT, though in this case it is an explicit part of the model specification. For AT-FA and the three DADT setups, we only considered words that were assigned to authors, rather than to fictitious authors or documents respectively.

Figure 4.4 demonstrates the potential advantage of DADT over LDA, AT and AT-FA: DADT allows us to explicitly specify how many topics we want to dedicate to authors and how many topics should be dedicated to documents. This allows the author representations yielded by DADT to be free from document words and to clearly exhibit the differences and similarities between authors. Of the other three models, the representation yielded by AT-FA is most similar to DADT's. However, AT-FA allocated Author 2's words to three topics, while Author 1's words were allocated only to two topics because AT-FA does not explicitly distinguish between

author topics and document topics. Similarly, while both LDA and AT capture differences and similarities between the authors, it appears that some topics are irrelevant to the goal of modelling authors, as topic 1 in LDA and topic 4 in AT were used with almost equal frequencies by both authors.

Summary

While the results presented in this section were obtained with a very simple synthetic dataset, they demonstrate the strengths and weaknesses of the models we considered:

- LDA seems suitable when we are interested only in document topics, as it fails to capture author topics.
- AT seems suitable when we are interested only in author topics, as it fails to capture document topics.
- AT-FA can capture both document topics and author topics, but it may underperform because both documents and authors are modelled over a single topic set.
- DADT addresses AT-FA's limitations by modelling documents and authors over two disjoint topic sets, and offers more flexibility than AT-FA in terms of the specification of priors. However, this flexibility may require practitioners to run more experiments to obtain the desired results.

It is worth noting that while our focus in this section was on the interpretability of the models, what ultimately determines the goodness of a model is its suitability for the task at hand. This is demonstrated throughout this thesis, where we use the above models for the three tasks that form the core of this study.

4.5 Future Outlook: Considering Word Order

The three models introduced in this chapter all rely on the assumption that words are drawn independently of each other (the "bag-of-words" assumption). While this assumption simplifies model inference, it can potentially cause loss of useful information. In this section, we discuss two possible approaches to considering word order: a preprocessing approach (Section 4.5.1) and a modelling approach (Section 4.5.2). Even though these approaches often yield topics that are easier to interpret than those yielded by bag-of-words models, preliminary experiments showed that they were of no benefit in terms of authorship attribution accuracy. Hence, we leave large-scale experiments with order-aware approaches for future work, and focus our attention throughout this thesis on bag-of-words methods.

4.5.1 Replacing Multi-word Expressions

A common way of incorporating order awareness into models that operate under the bag-of-words assumption is by adding a preprocessing step that detects all the

n-grams (e.g., bigrams and trigrams) in the documents and adds them to the documents as "words". This results in each document being a bag of n-grams rather than unigrams. The advantage of this approach is that it is easy to implement. However, there are several disadvantages:

- Adding all the n-grams potentially introduces noise, as many of them are likely to be irrelevant to the model's goal. For example, if we look at bigrams taken from this sentence ("*for example*", "*example ,*", "*, if*", "*if we*", etc.), only few exhibit any meaningful dependency between the tokens that constitute them.
- In large corpora, the number of possible n-grams can make model inference too computationally expensive.
- Using bags of n-grams with the topic models we considered can be problematic since it violates the modelling assumption of "word" independence. For example, the bigrams "*for example*" and "*example ,*" are strongly dependent, as is any pair of consecutive bigrams.

To address the above disadvantages while conserving the advantage of simplicity, we tested a different preprocessing approach that is based on research into detection of multi-word expressions (Sag et al., 2002). We used the jMWE toolkit (Kulkarni and Finlayson, 2011) to detect multi-word expressions, and *replaced* the constituent words with the multi-word expression in the document's representation.[10] For example, the fragment "*in new south wales this statute was in force*" from the Judgement dataset (Section 3.3.3) is converted to the bag-of-multi-word-expressions: "*in*", "*new_south_wales*", "*this*", "*statute*", "*was*", "*in_force*". Unlike the detection of n-grams, detection of multi-word expressions requires language-specific tools, but it addresses the disadvantages mentioned above:

- Only multi-word expressions with linguistic meaning are replaced.
- The number of "words" in this representation is lower than in the bag-of-words representation, making model inference faster.
- Since multi-word expressions replace their constituent words, the conditional independence assumption of the topic models is strengthened rather than weakened.

This approach also has the advantage of improved interpretability of topics. For example, in experiments on the Judgement dataset we found that topics that previously contained the words "*new*", "*south*", and "*wales*", now contain the multi-word expression "*new_south_wales*". Despite this advantage, we found that this approach had a negligible effect on authorship attribution performance in preliminary experiments. This is probably because most words do not take part in multi-word

[10] This approach was applied to topic modelling in the past, e.g., in Boyd-Graber et al.'s (2007) study on performing word sense disambiguation with an extended version of LDA.

expressions, and thus the introduction of multi-word expressions did not add much useful information. Hence, and in order to keep our results general and independent of the jMWE toolkit, we decided not to pursue this preprocessing direction for any of the tasks investigated in this thesis.

4.5.2 The HMM-LDA-AT Model

Several topic models that take word order into account have been suggested. As discussed in Section 2.2.3, while some of these models deal with word order by adding a large number of parameters, making it too computationally expensive to infer them on large corpora, others, such as the HMM-LDA model (Griffiths et al., 2004), handle word order in a way that is less computationally intensive. We implemented HMM-LDA-AT, which is an extension to Griffiths et al.'s (2004) model that combines HMM-LDA with the AT model. We describe HMM-LDA-AT below, but we did not use it for other experiments in this thesis because, as mentioned in the beginning of this section, preliminary experiments showed that HMM-LDA-AT had no advantage over DADT on the authorship attribution task.

HMM-LDA

The main idea behind HMM-LDA is to combine HMM and LDA so that the HMM component captures the short-term dependencies between words (i.e., on the sentence level), while the LDA component models the long-term dependencies (i.e., on the document and corpus level).

HMM-LDA dedicates one class of the HMM to the LDA component, while the other classes are associated with simple distributions over words, as in standard HMM. One advantage of this model over LDA is that content words tend to be associated with the class that is dedicated to the LDA component, while the other words tend to be allocated to the HMM classes that are not associated with the LDA component, with each class's words belonging to a different part of speech. This obviates the need to filter out stopwords in a preprocessing step (when using LDA for its traditional purpose of finding human-interpretable document topics). More importantly from our perspective, HMM-LDA may enable user modelling according to similar reasoning to that behind our DADT model: separating document words (as represented by the LDA component) from author words (as represented by the other HMM classes) may have benefits in terms of user modelling.

The generative procedure of HMM-LDA is described formally in (Griffiths et al., 2004). To avoid introducing more notation, we provide an informal description here:

- *Corpus level.*
  - For each class:
    - Draw a distribution over classes that indicates the transition probabilities to each class.

        * If the class number is not 1 (i.e., it is a non-LDA class), draw the class's distribution over words.
    - For each document topic, draw a distribution over words (as in LDA).
- *Document level.* For each document in the corpus:
    - Draw the document's distribution over document topics (as in LDA).
    - *Word level.* For each word in the document:
        * Draw the word's class from the transition distribution of the previous word's class.
        * If the word's class number is 1 generate the word as in LDA:
            · Draw the word's topic from the document's distribution over document topics.
            · Draw the word from its topic's word distribution.
        * Otherwise, draw the word from its class's distribution over words.

HMM-LDA-AT

The main idea behind our HMM-LDA-AT model is to combine HMM-LDA with AT by drawing the words in the non-LDA HMM classes according to the authors, as in AT. This requires adjusting HMM-LDA's generative procedure as follows (the changes are highlighted in blue):[11]

- *Corpus level.*
    - For each class:
        * Draw a distribution over classes that indicates the transition probabilities to each class.
        * If the class number is not 1 (i.e., it is a non-LDA class), draw distributions as in AT:
            · For each author, draw a class-specific distribution over author topics.
            · For each class-specific author topic, draw a distribution over words.
    - For each document topic, draw a distribution over words (as in LDA).
- *Document level.* For each document in the corpus:
    - Draw the document's distribution over document topics (as in LDA).
    - *Word level.* For each word in the document:

[11] For simplicity, we assume that all the texts are single-authored and that the authors are observed.

- Draw the word's class from the transition distribution of the previous word's class.
- If the word's class number is 1 generate the word as in LDA:
  - Draw the word's topic from the document's distribution over document topics.
  - Draw the word from its document topic's word distribution.
- Otherwise, generate the word as in AT:
  - Draw the word's topic from the observed author's class-specific distribution over author topics.
  - Draw the word from its topic's class-specific author word distribution.

It is worth noting that the above description assumes that all non-LDA classes are AT classes. However, the model can be easily extended to allow for a flexible number of AT classes, i.e., some non-LDA classes can be associated with an AT model while others would be associated with a distribution over words. This would allow words whose usage does not vary between authors and documents to be allocated to non-LDA, non-AT classes, while words that differentiate authors would get allocated to the AT classes. Similarly, it is easy to extend the model to include more than one LDA class.[12]

Preliminary Experimentation

We implemented both HMM-LDA and HMM-LDA-AT using collapsed Gibbs sampling, and ran experiments on the Judgement dataset with ten classes, ten document topics and two author topics per AT class. We found that we could reproduce Griffiths et al.'s (2004) HMM-LDA results: content words were allocated to the LDA class, while function words and corpus-specific stopwords were allocated to the other classes. For example, the top words in one LDA topic were *will*, *death*, and *estate*, while virtually all occurrences of the articles *the*, *a*, and *an* were allocated to a single non-LDA class, and nouns like *case* and *court* were allocated to another non-LDA class (these nouns can be seen as corpus-specific stopwords because they appear in most judgements).[13] When we ran HMM-LDA-AT, the LDA topics remained virtually the same, and each of the non-LDA classes was split into two author topics. These author topics are harder to interpret, but we observed that in some AT

[12]This may result, for example, in the allocation of nouns that vary across documents to one LDA class, while verbs that vary across documents would be allocated to a different LDA class.

[13]When running DADT on the Judgement dataset, we obtained a similar separation of content and non-content words into document and author topics respectively, probably because all the judgements share common themes and none of the judges in the dataset specialised in cases of a specific type.

classes the authors' distributions over topics were virtually the same, while in other AT classes these distributions varied from author to author.

When we moved on to test the performance of HMM-LDA-AT on authorship attribution, we found that it did not outperform DADT, even when we varied the number of classes and topics, and extended the model to include both AT classes and non-AT, non-LDA classes. This may be because we could not find the correct set of configurable parameters, which is due to the number of parameters that need to be set in advance in HMM-LDA-AT in addition to prior values: the overall number of classes, the number of AT classes, the number of document topics, and the number of author topics. Hence, we believe that an extension that reduces the number of configurable parameters is required to make HMM-LDA-AT feasible for practical use and experimentation. One possible direction is adapting Teh et al.'s (2006) procedure for learning the number of topics in LDA to perform topic and class number inference in HMM-LDA-AT. We leave the implementation of such an extension for future work.

4.6 Summary and Conclusions

In this chapter, we discussed the application of topic models to modelling users who authored texts of any type. We described two existing topic models – *Latent Dirichlet Allocation* (LDA) and the *Author-Topic* (AT) model – and showed how they can be applied to user modelling. In addition, we introduced the *Disjoint Author-Document Topic* (DADT) model, which combines LDA and AT into a single model. We compared the interpretability of the topics obtained by the three models by running experiments on a synthetic dataset, and suggested ways of taking word order into account in the context of topical user modelling.

The models presented in this chapter can be applied to many user modelling scenarios where user-generated texts are available, going beyond those discussed in this thesis. For example, any system that adapts its behaviour to individual users can potentially benefit from compact representations of users as topic distributions, which may be considered as additional features that describe the users. In subsequent chapters we explore the application of these models to the three tasks we consider in this thesis, as outlined at the beginning of this chapter. Specifically, in Chapter 5 we explore the application of topical user models to authorship attribution, in Chapter 6 we employ topical user models to measure user similarity in our user-based polarity inference framework, and in Chapter 7 we use topical user models to obtain low-dimensional representations of users for our rating prediction framework.

Chapter 5

Authorship Attribution with Topical User Models

This chapter investigates the application of the topical user models from Chapter 4 to the authorship attribution task. As mentioned in Section 2.3, approaches that are applicable to authorship attribution, where the goal is to identify the authors of anonymous texts, are often also useful for authorship profiling, which deals with inferring user characteristics (e.g., demographic information and personality traits) from texts. Our main goal in this chapter is to gauge whether topical user models retain information from user-generated texts that is representative of user characteristics. This is done through authorship attribution experiments.

Generally, our topic-based authorship attribution methods consist of the following steps:

1. Given *training texts* with known authors, train a topic model following the model-specific procedure outlined in Chapter 4. The models we consider in this chapter are LDA, AT, AT with fictitious authors (AT-FA), and DADT.
2. Given an anonymous *test text*, infer its topic distributions according to the model-specific procedure, as described in Section 5.1.
3. Assign the test text to one of the candidate authors by employing one of the methods described in Section 5.2. Each one of these methods can be categorised as belonging to one of three approaches: (1) *dimensionality reduction*, where the topic distributions are used as input to another classifier; (2) *distance-based*, where the distance between the test text's distributions and the training distributions is used to find the closest author; and (3) *probabilistic*, where the probabilistic structure of the underlying model is used to find the most probable author of the test text.

Our evaluation shows that the best topic-based methods yield state-of-the-art performance in several scenarios where the number of candidate authors ranges from

three to about 20,000, with the best results in most cases obtained by methods based on our DADT model (Section 5.3). This suggests that topical user models can capture user characteristics as reflected by user-generated texts due to the strong correlation between authorship attribution and profiling performance – a conclusion that we rely on in our use of topic models in subsequent chapters.

While our main focus in this thesis is on single-authored texts, AT and DADT can also be used to model authors based on multi-authored texts, such as research papers. To demonstrate the potential utility of this capability of the models, we present the results of a preliminary study, where we use AT and DADT to identify anonymous reviewers based on publicly-available information (reviewer lists and the reviewers' publications, which are often multi-authored). Our results indicate that reviewers may be identified with moderate accuracy, at least in small conference tracks and workshops. We hope that these results will help fuel the ongoing discussion in the research community on addressing anonymity (see, e.g., Daumé III, 2012).

This chapter is structured as follows. Section 5.1 presents the procedures we use to infer the topic distributions of test texts. Section 5.2 introduces our topic-based authorship attribution methods, which are evaluated in Section 5.3. Section 5.4 discusses our reviewer identification experiments, and Section 5.5 concludes the chapter.

5.1 Topic Inference for Unseen Documents

In this chapter, we consider the closed-set authorship attribution task, i.e., training texts by the candidate authors are supplied in advance, and for each test text, the goal is to attribute the text to the correct author out of the candidate authors (Section 2.3.1). We consider only the case where test texts are given one by one, making this a fully-supervised classification problem (in contrast to semi-supervised classification, where the algorithm is given a set of test texts). Hence, the topic-based methods we introduce in this chapter work in two phases: (1) *training*, where the underlying topic model is inferred following the procedures described in Chapter 4; and (2) *classification*, where the author of the test text is found. Most of the topic-based methods we introduce in Section 5.2 require a way of inferring the topic distributions of previously-unseen documents, i.e., the test texts. In this section, we describe the procedures we employ to infer these distributions for each of the topic models: LDA, AT, AT-FA, and DADT.

5.1.1 LDA

As discussed in Section 4.2.2, we follow a Gibbs sampling approach to infer LDA from training texts, where several samples are retained from each sampling chain.

For each retained training sample, we set the topic distribution $\boldsymbol{\theta}_d^{(D)}$ of each document d and the word distribution $\boldsymbol{\phi}_t^{(D)}$ of each document topic t to their expected values (Table 4.1).

In the classification phase, we are given a test text $\tilde{\boldsymbol{w}}$ (a word vector of length $\tilde{N}$). For each training sample, we infer the topic distribution $\tilde{\boldsymbol{\theta}}^{(D)}$ of the test text by running Gibbs sampling, where the word distributions $\boldsymbol{\phi}_t^{(D)}$ are given as their expected values according to the training sample. This involves repeatedly sampling from:[1]

$$p\left(\tilde{z}_i = t|\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{z}}_{-i}; \boldsymbol{\Phi}^{(D)}, \boldsymbol{\alpha}^{(D)}\right) \propto \frac{\alpha_t^{(D)} + \tilde{c}_t^{(DT)}}{\sum_{t'=1}^{T^{(D)}} \left(\alpha_{t'}^{(D)} + \tilde{c}_{t'}^{(DT)}\right)} \phi_{t\tilde{w}_i}^{(D)} \tag{5.1}$$

where $\tilde{z}_i$ is the topic assignment for the i-th word in $\tilde{\boldsymbol{w}}$, $\tilde{\boldsymbol{z}}_{-i}$ contains all of $\tilde{\boldsymbol{w}}$'s topic assignments except for the i-th assignment, and $\tilde{c}_t^{(DT)}$ is the count of words assigned to topic t, excluding the i-th assignment.

As done in the training phase, we set $\tilde{\boldsymbol{\theta}}^{(D)}$ to its expected value according to:

$$\mathbb{E}[\tilde{\theta}_t^{(D)}] = \frac{\alpha_t^{(D)} + \tilde{c}_t^{(DT)}}{\sum_{t'=1}^{T^{(D)}} \left(\alpha_{t'}^{(D)} + \tilde{c}_{t'}^{(DT)}\right)} \tag{5.2}$$

where $\tilde{c}_t^{(DT)}$ now contains the counts over the full vector of topic assignments $\tilde{\boldsymbol{z}}$. Note that since we assume that the $\boldsymbol{\phi}_t^{(D)}$ values are given at the classification phase, the topics are *not* exchangeable. This means that we can average the $\mathbb{E}[\tilde{\theta}_t^{(D)}]$ values across test samples obtained from the same sampling chain.

5.1.2 AT

In a similar manner to LDA, after running Gibbs sampling to infer the AT model, we obtain several training samples (Section 4.2.2). The expected distributions obtained from each AT training sample are the topic distribution $\boldsymbol{\theta}_a^{(A)}$ of each author a and the word distribution $\boldsymbol{\phi}_t^{(A)}$ of each author topic t (Table 4.1).

In the classification phase, we do not know the author $\tilde{a}$ of the test text $\tilde{\boldsymbol{w}}$ (we assume that test texts are single-authored). If we knew the real author, no sampling would be required to obtain the author's topic distribution because it is already inferred in the training phase ($\boldsymbol{\theta}_a^{(A)}$ above). However, since we do not know who the author is, we assume that $\tilde{a}$ is a "new", previously-unknown author, and employ Gibbs sampling to infer this author's topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ by repeatedly sampling from:

$$p\left(\tilde{z}_i = t|\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{z}}_{-i}; \boldsymbol{\Phi}^{(A)}, \boldsymbol{\alpha}^{(A)}\right) \propto \frac{\alpha_t^{(A)} + \tilde{c}_t^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + \tilde{c}_{t'}^{(AT)}\right)} \phi_{t\tilde{w}_i}^{(A)} \tag{5.3}$$

[1] Note that this is a simplified version of Equation 4.3 where the word distributions are known and sampling is performed only for one document.

where $\tilde{z}_i$ is the topic assignment for the i-th word in $\tilde{\boldsymbol{w}}$, $\tilde{\boldsymbol{z}}_{-i}$ contains all of $\tilde{\boldsymbol{w}}$'s topic assignments except for the i-th assignment, and $\tilde{c}_t^{(AT)}$ is the count of topic t assignments to author $\tilde{a}$, excluding the i-th assignment.

Similarly to LDA, we set $\tilde{\boldsymbol{\theta}}^{(A)}$ to its expected value according to:

$$\mathbb{E}[\tilde{\theta}_t^{(A)}] = \frac{\alpha_t^{(A)} + \tilde{c}_t^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + \tilde{c}_{t'}^{(AT)}\right)} \tag{5.4}$$

where $\tilde{c}_t^{(AT)}$ now contains the counts over the full vector of author topic assignments $\tilde{\boldsymbol{z}}$.

5.1.3 AT-FA

When AT is run with an additional fictitious author (FA) for each of the training texts, it yields a topic distribution $\boldsymbol{\theta}_a^{(A)}$ for each author a and the word distribution $\boldsymbol{\phi}_t^{(A)}$ of each author topic t (as in AT). In addition, AT-FA yields an *author* topic distribution $\boldsymbol{\theta}_d^{(A)}$ for each document d – these distributions are over the same topic set as the author topic distributions, since documents are represented by fictitious authors.

In the classification phase, as we do not know the real author of the test text $\tilde{\boldsymbol{w}}$, we cannot assume that the test text was co-written by a previously-unknown author $\tilde{a}$ and a fictitious author. Making this assumption would require us to infer the topic distributions for *two* previously-unknown authors – the fictitious author and the real author – with no way of telling them apart. Therefore, we consider two alternatives to this assumption:

1. Assume that the test text was written only by a real author (without a fictitious author), and follow AT's inference procedure, as described in Section 5.1.2.
2. For each training author a, assume that the test text was written by a together with a fictitious author f_a, and infer the topic distribution of f_a as described below.

It is worth noting that the second alternative is not applicable to all the methods based on AT-FA (Section 5.2.4). Further, as the second alternative requires running a separate sampling procedure for each of the candidate authors, it becomes too computationally expensive to run in scenarios with many candidate authors and test texts. In addition, our DADT model offers a less ad-hoc solution to this issue, as both authors and documents are represented by DADT, and thus it allows us to infer the test text's document topic distribution together with an author topic distribution for the test text's previously-unknown author (Section 5.1.4).

The inference procedure when following the second alternative, where we assume that the test text was written by a known author a together with a fictitious

author f_a, involves repeatedly sampling from:

$$p\left(\tilde{x}_i = x, \tilde{z}_i = t | \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{x}}_{-i}, \tilde{\boldsymbol{z}}_{-i}; \boldsymbol{\Phi}^{(A)}, \boldsymbol{\Theta}^{(A)}, \boldsymbol{\alpha}^{(A)}\right) \propto \tag{5.5}$$
$$\begin{cases} \dfrac{\alpha_t^{(A)} + \tilde{c}_{f_a t}^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + \tilde{c}_{f_a t'}^{(AT)}\right)} \phi_{t\tilde{w}_i}^{(A)} & \text{if } x = f_a \\ \theta_{at}^{(A)} \phi_{t\tilde{w}_i}^{(A)} & \text{if } x = a \end{cases}$$

where $\tilde{x}_i$ is the author assignment for the i-th word in $\tilde{\boldsymbol{w}}$, $\tilde{\boldsymbol{x}}_{-i}$ contains all of $\tilde{\boldsymbol{w}}$'s author assignments except for the i-th assignment, and $\tilde{c}_{f_a t}^{(AT)}$ is the count of topic t assignments to the fictitious author f_a, excluding the i-th assignment (the other variables are defined as in Equation 5.3).

Similarly to AT, we set the fictitious author's topic distribution $\tilde{\boldsymbol{\theta}}_{f_a}^{(A)}$ to its expected value according to:

$$\mathbb{E}[\tilde{\theta}_{f_a t}^{(A)}] = \frac{\alpha_t^{(A)} + \tilde{c}_{f_a t}^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + \tilde{c}_{f_a t'}^{(AT)}\right)} \tag{5.6}$$

where $\tilde{c}_{f_a t}^{(AT)}$ now contains the counts over the full vectors of author and topic assignments ($\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{z}}$ respectively).

5.1.4 DADT

After the training phase of DADT, we obtain the expected values of the document topic distribution $\boldsymbol{\theta}_d^{(D)}$ of each document d, the word distribution $\boldsymbol{\phi}_t^{(D)}$ of each document topic t, the author topic distribution $\boldsymbol{\theta}_a^{(A)}$ of each author a, the word distribution $\boldsymbol{\phi}_t^{(A)}$ of each author topic t, the author/document topic ratio π_d of each document d, and the corpus distribution over authors $\boldsymbol{\chi}$ (Tables 4.1 and 4.2).

In the classification phase, we do not know the author $\tilde{a}$ of the test text $\tilde{\boldsymbol{w}}$. As in AT, we assume that $\tilde{a}$ is a previously-unknown author. Then, we infer $\tilde{a}$'s author topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ together with the test text's document topic distribution $\tilde{\boldsymbol{\theta}}^{(D)}$ and author/document topic ratio $\tilde{\pi}$ by repeatedly sampling from:

$$p\left(\tilde{y}_i = y, \tilde{z}_i = t | \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{y}}_{-i}, \tilde{\boldsymbol{z}}_{-i}; \boldsymbol{\Phi}^{(D)}, \boldsymbol{\Phi}^{(A)}, \boldsymbol{\alpha}^{(D)}, \boldsymbol{\alpha}^{(A)}, \delta^{(D)}, \delta^{(A)}\right) \propto \tag{5.7}$$
$$\begin{cases} \left(\delta^{(D)} + \tilde{c}^{(DD)}\right) \dfrac{\alpha_t^{(D)} + \tilde{c}_t^{(DT)}}{\sum_{t'=1}^{T^{(D)}} \left(\alpha_{t'}^{(D)} + \tilde{c}_{t'}^{(DT)}\right)} \phi_{t\tilde{w}_i}^{(D)} & \text{if } y = 0 \\ \left(\delta^{(A)} + \tilde{c}^{(DA)}\right) \dfrac{\alpha_t^{(A)} + \tilde{c}_t^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + \tilde{c}_{t'}^{(AT)}\right)} \phi_{t\tilde{w}_i}^{(A)} & \text{if } y = 1 \end{cases}$$

where $\tilde{y}_i$ is the topic indicator for the i-th word, $\tilde{\boldsymbol{y}}_{-i}$ contains all of $\tilde{\boldsymbol{w}}$'s topic indicators except for the i-th indicator, and $\tilde{c}^{(DD)}$ and $\tilde{c}^{(DA)}$ are the counts of words assigned to document and author topics respectively, excluding the i-th assignment (the other variables are defined as in Equations 5.1 and 5.3).

The expected values of $\tilde{\boldsymbol{\theta}}^{(D)}$ and $\tilde{\boldsymbol{\theta}}^{(A)}$ are the same as for LDA and AT (Equations 5.2 and 5.4 respectively). The expected value of $\tilde{\pi}$ is:

$$\mathbb{E}[\tilde{\pi}] = \frac{\delta^{(A)} + \tilde{c}^{(DA)}}{\delta^{(D)} + \delta^{(A)} + \tilde{N}} \tag{5.8}$$

where $\tilde{c}^{(DA)}$ now contains the counts over the full vector of indicators $\tilde{\boldsymbol{y}}$.

In an analogous manner to the second AT-FA alternative (Section 5.1.3), if we were only interested in the test text's document topic distribution $\tilde{\boldsymbol{\theta}}^{(D)}$ and author/document topic ratio $\tilde{\pi}$, we could perform a separate inference procedure for each author a by replacing $\frac{\alpha_t^{(A)} + \tilde{c}_t^{(AT)}}{\sum_{t'=1}^{T^{(A)}} \left(\alpha_{t'}^{(A)} + \tilde{c}_{t'}^{(AT)}\right)}$ with $\theta_{at}^{(A)}$ in Equation 5.7. However, employing this approach is too computationally expensive in scenarios with many candidate authors. In addition, we found in preliminary experiments that methods based on separate sampling yield comparable results to methods based on the assumption that the test texts were written by a previously-unknown author (Section 5.2.5). Hence, we only report results obtained when following this assumption.

5.2 Authorship Attribution Methods

This section introduces the authorship attribution methods considered in this chapter. In Section 5.2.1, we briefly discuss our baseline method (SVMs trained on tokens), while Sections 5.2.2, 5.2.3, 5.2.4 and 5.2.5 introduce methods based on LDA, AT, AT-FA and DADT respectively.

We consider three approaches to employing topical user models in authorship attribution: dimensionality reduction, distance-based and probabilistic.

Under the *dimensionality reduction* approach, the original documents are converted to topic distributions, and the topic distributions are used as input to a classifier. Generally, this approach makes it possible to use classifiers that are too computationally expensive to employ with a large feature set, e.g., Webb et al.'s (2005) AODE classifier, whose time complexity is quadratic in the number of features. In our case, we use the reduced document representations as input to SVMs, and compare their performance to the performance obtained with SVMs trained directly on tokens (denoted *Token SVMs*). This allows us to roughly gauge how much information is lost by converting texts from token representations to topic representations. However, employing a classifier on top of the topic model does not fully test the utility of the author representations yielded by the model – these are better tested by the next two approaches.

Distance-based methods employ a distance measure to find the author whose topic distributions are closest to the distributions inferred from the test text. We use the Hellinger distance, which is defined as:[2]

$$\mathrm{H}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \sqrt{\frac{1}{2} \sum_{t=1}^{T} \left(\sqrt{\theta_{1t}} - \sqrt{\theta_{2t}} \right)^2} \tag{5.9}$$

where $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are two T-dimensional categorical distributions. This allows us to directly test how representative of the authors are the inferred topic distributions (as specified for each topic model in the remainder of this section). However, this approach may perform poorly because it does not fully consider the underlying structure of the topic model. Instead, it utilises the given topic distributions but treats the rest of the model as a black box.

In contrast to distance-based methods, *probabilistic* methods employ the underlying model's definitions directly to estimate the probability that a given author wrote a given test text. These methods require the model to be aware of authors, which means that LDA cannot be used in this case. We expect this approach to yield the best performance because unlike the dimensionality reduction and distance-based approaches, the probabilistic approach considers the structure of the topic model. Further, the models are trained to maximise the probability of the inferred parameters rather than the distance between authors, which, in general, should allow the probabilistic approach to outperform the other two approaches.

5.2.1 Baseline: Token SVMs

Our baseline method is SVMs trained on token frequency features (i.e., token counts divided by the total number of tokens in the document), which are known to yield good authorship attribution performance (Section 2.3.3). When multiple authors exist, we employ the one-versus-all setup. This setup scales linearly in the number of authors and was shown to be at least as effective as other multi-class SVM approaches in many cases (Rifkin and Klautau, 2004).

It is worth noting that unlike the topic models, the Token SVM baseline is trained with the goal of maximising the authorship attribution accuracy, which may give Token SVM an advantage over topic-based methods. Further, as SVMs are discriminative classifiers, they may yield better performance than probabilistic topic-based methods, which can be seen as generative classifiers (Ng and Jordan, 2001). However, as demonstrated by Ng and Jordan's comparison of discriminative and

[2]It is worth noting that we considered other measures for comparing topic distributions, including Kullback-Leibler divergence and Bhattacharyya distance. We chose the Hellinger distance because it satisfies all the required properties of a distance metric (non-negativity, identity of indiscernibles, symmetry and triangle inequality).

generative classifiers, this better performance may only be obtained in the presence of "enough" training data (just how much data is "enough" depends on the dataset).

5.2.2 Methods Based on LDA

Dimensionality Reduction: LDA-SVM. Employing LDA for dimensionality reduction is relatively straightforward – all it entails is converting the training and test texts to topic distributions as described in Sections 4.2.2 and 5.1.1 respectively, and using these topic distributions as classifier features. As mentioned above, the classifier we use is SVM, which makes it possible to directly compare the results obtained with the LDA-SVM method to the baseline results obtained by running SVMs trained directly on token frequencies.

This LDA-SVM approach was employed by Blei et al. (2003) to demonstrate the dimensionality reduction capabilities of LDA on the task of classifying articles according to a set of predefined categories. To the best of our knowledge, only Rajkumar et al. (2009) have previously applied LDA-SVM to authorship attribution – they published preliminary results obtained by running LDA-SVM, but did not compare their results to a Token SVM baseline. In Section 5.3, we present the results of more extensive experiments on the applicability of this approach to authorship attribution.

Distance-based: LDA-H. The LDA-H method measures the Hellinger distance between the test text's topic distribution $\tilde{\boldsymbol{\theta}}^{(D)}$ and the topic distributions $\boldsymbol{\theta}_d^{(D)}$ of each training document d, and returns the author with the lowest mean distance to all of his/her documents. Formally, LDA-H returns:

$$\underset{a\in\{1,\ldots,A\}}{\arg\min} \frac{1}{|\mathcal{D}_a|} \sum_{d\in\mathcal{D}_a} \mathrm{H}\left(\tilde{\boldsymbol{\theta}}^{(D)}, \boldsymbol{\theta}_d^{(D)}\right) \tag{5.10}$$

where $\mathcal{D}_a$ is the set of author a's training documents.

We use the mean distance to all the documents written by each author, rather than, e.g., taking a nearest-neighbour approach on the document level (where the author of the nearest document is returned), because we want to measure how well LDA represents the authors as a whole (i.e., as sets of topic distributions, as discussed in Section 4.2.3). Nonetheless, under LDA-H the differences between individual training documents by each author have some indirect influence on the results, especially for authors who wrote only a few documents, where each document has more effect on the mean distance than for authors with many training documents.

5.2.3 Methods Based on AT

Dimensionality Reduction: AT-SVM. We cannot use AT to obtain *document* topic distributions, since AT only infers *author* topic distributions (Section 4.2).

Hence, we train the SVM component on the author topic representations (each document is converted to its author topic distribution). For each test text, we infer the author topic distribution under the assumption that the test text was written by a previously-unknown author (Section 5.1.2), and then classify this distribution. This may be seen as very radical dimensionality reduction, since each author's entire set of training documents is reduced to a single author topic distribution.

Distance-based: AT-H. This method assumes that the test document was written by a previously-unknown author, and infers this author's topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ (Section 5.1.2). It then attributes the document to the training author a whose topic distribution $\boldsymbol{\theta}_a^{(A)}$ is the closest to $\tilde{\boldsymbol{\theta}}^{(A)}$. Formally, AT-H returns:

$$\arg\min_{a\in\{1,\dots,A\}} \mathrm{H}\left(\tilde{\boldsymbol{\theta}}^{(A)}, \boldsymbol{\theta}_a^{(A)}\right) \tag{5.11}$$

AT-H can be seen as an extreme version of LDA-H, where each author's set of training distributions is collapsed into a single distribution. An advantage of AT-H over LDA-H is that AT-H scales linearly in the number of authors (for each test text, the Hellinger distance is measured only once for each candidate author), while LDA-H scales linearly in the number of documents, as distance is measured to each training document. However, our main focus is on improving classification accuracy, and LDA-H may prove to be more accurate in some cases as it takes all the training documents into account as separate entities. On the other hand, using AT-H may have a de-noising effect since AT is expected to generate author topic distributions that are highly probable according to the training corpus, while LDA aims to obtain document topic distributions without being aware of the existence of authors.

Probabilistic: AT-P. For each author a, AT-P calculates the probability of the test text words given the AT model inferred from the training texts, under the assumption that the test text was written by a. It returns the author for whom this probability is the highest:

$$\arg\max_{a\in\{1,\dots,A\}} p\left(\tilde{\boldsymbol{w}}|\tilde{a}=a, \boldsymbol{\Theta}^{(A)}, \boldsymbol{\Phi}^{(A)}\right) \propto \arg\max_{a\in\{1,\dots,A\}} \prod_{i=1}^{\tilde{N}} \sum_{t=1}^{T^{(A)}} \theta_{at}^{(A)} \phi_{t\tilde{w}_i}^{(A)} \tag{5.12}$$

This method does not require any topic inference in the classification phase, because the author topic distributions $\boldsymbol{\Theta}^{(A)}$ and topic word distributions $\boldsymbol{\Phi}^{(A)}$ are already inferred at training time. It is worth noting that in practice we use the log of the above probability for reasons of numerical stability, but this is mathematically equivalent to using the probability directly.

As mentioned at the beginning of this section, we expect AT-P to outperform AT-SVM and AT-H since AT-P relies directly on the probabilistic structure of the

AT model. In addition, AT-P has the advantage of not requiring any topic inference in the classification phase.

It is worth noting that we also performed preliminary experiments with a method that: (1) assumes that the test text was co-written by all the candidate authors, (2) infers the word-to-author assignments for the test text, and (3) returns the author that was attributed the most words. However, we found that this method yields poor results in comparison to other AT-based approaches in three-way authorship attribution. In addition, it proved too computationally expensive to run this method in cases with many candidate authors, as it requires iterating through all the authors for every test text word in each sampling iteration.

5.2.4 Methods Based on AT-FA

AT-FA is the same model as AT, but it is run with the preprocessing step of adding an additional fictitious author to each training document. However, different constraints apply to AT-FA in the classification phase. This is because in this phase, we cannot conserve AT-FA's assumption that all the texts are written by a real author together with a fictitious author, since we do not know who wrote the test text (Section 5.1.3). Hence, if we were to assume that the real author is a previously-unknown author, as done for AT, we would have no way of telling the previously-unknown author from the fictitious author, since they are both unique to the test text. In Section 5.1.3 we suggested two possible ways of addressing this issue:

1. Assume that the test text was written only by a real, previously-unknown, author (without a fictitious author), and infer this author's topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ (as in AT).

2. For each training author a, assume that the test text was written by a together with a fictitious author f_a and infer the fictitious author's topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}_{f_a}$. This results in a set of fictitious author topic distributions, each matching a training author.

While the second alternative may appear more attractive because it does not violate the fictitious author assumption of AT-FA, we cannot use it with the dimensionality reduction and distance-based methods (AT-FA-SVM and AF-FA-H respectively, as described below), as these methods require inferring the topic distribution of the previously-unknown author $\tilde{\boldsymbol{\theta}}^{(A)}$.

Dimensionality Reduction: AT-FA-SVM. AT-FA yields a topic distribution for each training document (i.e., the topic distribution of the fictitious author associated with the document), and a topic distribution for each real author (all the distributions are over the same topic set). We convert each training document to the concatenation of these two distributions, and use this concatenation as input to the SVM component. In the classification phase, we assume that the test text was

written by a single previously-unknown author, and represent the test text as the concatenation of the inferred topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ to itself.

It is worth noting that our DADT model offers a more elegant solution than concatenating the same distribution to itself, because DADT differentiates between author topics and document topics – a distinction that AT-FA attempts to capture through fictitious authors. Hence, we expect the DADT-SVM approach, which we define in Section 5.2.5, to perform better than AT-FA-SVM. Nonetheless, we also experiment with AT-FA-SVM for the sake of completeness.

Distance-based: AT-FA-H. For the distance-based method, we also follow the first alternative of assuming that the test text was written only by a previously-unknown author (without a fictitious author). Hence, AT-FA-H is identical to AT-H in the classification phase: it infers the previously-unknown author's topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ and returns the (real) training author whose topic distribution is the closest to $\tilde{\boldsymbol{\theta}}^{(A)}$ (Equation 5.11).

Probabilistic: AT-FA-P. For the probabilistic approach, we consider two variants, matching the two alternatives outlined above:

1. **AT-FA-P1.** This variant is identical in the classification phase to AT-P – it returns the author that maximises the probability of the test text's words according to Equation 5.12, assuming that the test text was not co-written by a fictitious author.

2. **AT-FA-P2.** This variant performs the following steps for each author a: (1) assume that the test text was written by a together with a fictitious author f_a; (2) infer the topic distribution of the fictitious author $\tilde{\boldsymbol{\theta}}^{(A)}_{f_a}$ (Section 5.1.3); (3) calculate the probability of the test text words under the assumption that it was co-written by a and f_a, and given the inferred $\tilde{\boldsymbol{\theta}}^{(A)}_{f_a}$; and (4) return the author for which the probability of the test text words is maximised:

$$\begin{aligned} &\arg\max_{a \in \{1,\dots,A\}} p\left(\tilde{\boldsymbol{w}} | \tilde{\boldsymbol{a}} = \{a, f_a\}, \tilde{\boldsymbol{\theta}}^{(A)}_{f_a}, \boldsymbol{\Theta}^{(A)}, \boldsymbol{\Phi}^{(A)}\right) \propto \\ &\arg\max_{a \in \{1,\dots,A\}} \prod_{i=1}^{\tilde{N}} \sum_{t=1}^{T^{(A)}} \left(\theta^{(A)}_{at} \phi^{(A)}_{t\tilde{w}_i} + \tilde{\theta}^{(A)}_{f_a t} \phi^{(A)}_{t\tilde{w}_i}\right) \end{aligned} \tag{5.13}$$

   where $\tilde{\boldsymbol{a}}$ is the test text's set of authors.

   The problem with this approach is that it is too computationally expensive to use on datasets with many candidate authors, as it requires running a separate inference procedure for each author. Nonetheless, in cases where AT-FA-P2 can be run, we expect it to perform better than AT-FA-P1 because it does not violate the fictitious author assumption of AT-FA.

5.2.5 Methods Based on DADT

Dimensionality Reduction: DADT-SVM. DADT yields a document topic distribution $\boldsymbol{\theta}_d^{(D)}$ for each document d, and an author topic distribution $\boldsymbol{\theta}_a^{(A)}$ for each author a. Similarly to AT-FA-SVM, we convert each training document to the concatenation of these two distributions, and use this concatenation as input to the SVM component.

In contrast to AT-FA, DADT's document topic distributions are defined over a topic set that is disjoint from the author topic set. This makes it possible to assume that the test text was written by a previously-unknown author, and obtain the test text's document distribution $\tilde{\boldsymbol{\theta}}^{(D)}$ together with the previously-unknown author's topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ (following the procedure described in Section 5.1.4). As in the training phase, test texts are represented as the concatenation of these two distributions.

We expect DADT-SVM to outperform AT-FA-SVM, since we are able to maintain the assumptions of DADT in the classification phase, which we cannot do in AT-FA-SVM. Further, DADT-SVM should perform better than AT-SVM, because DADT-SVM accounts for differences between individual documents while AT-SVM represents each author using a single training instance. Hypothesising about the expected performance of DADT-SVM in comparison to LDA-SVM is harder: we expect performance to be corpus-dependent to a certain degree – in datasets where differences between individual documents are important (e.g., with few authors who wrote many texts), LDA-SVM may have an advantage, as all the words are allocated to document topics. On the other hand, in datasets where the differences between authors are more important (e.g., with many authors who wrote a few texts), DADT-SVM may outperform LDA-SVM because it represents the authors explicitly.

Distance-based: DADT-H. We consider three different variants of the distance-based approach:

1. *Distance to Document Topics (DADT-HD).* Like LDA-H, this variant returns the author a that minimises the mean distance between the test text's document topic distribution $\tilde{\boldsymbol{\theta}}^{(D)}$ and the distributions over document topics of a's training texts (Equation 5.10).

   Note that we do not expect this variant to perform well, since document topics are expected to be representative of documents rather than authors, due to the disjoint nature of DADT.

2. *Distance to Author Topics (DADT-HA).* Like AT-H, this variant returns the author a that minimises the distance between the test text's author topic distribution $\tilde{\boldsymbol{\theta}}^{(A)}$ and a's distribution over author topics $\boldsymbol{\theta}_a^{(A)}$ (Equation 5.11).

This variant is expected to perform better than DADT-HD, since author topics should be representative of authors.

3. *Distance to Document and Author Topics (DADT-HDA).* This variant returns the author a that minimises the sum of the distances employed by DADT-HD and DADT-HA:

$$\underset{a\in\{1,\ldots,A\}}{\arg\min}\left(\frac{1}{|\mathcal{D}_a|}\sum_{d\in\mathcal{D}_a}\mathrm{H}\left(\tilde{\boldsymbol{\theta}}^{(D)},\boldsymbol{\theta}_d^{(D)}\right)\right)+\mathrm{H}\left(\tilde{\boldsymbol{\theta}}^{(A)},\boldsymbol{\theta}_a^{(A)}\right) \tag{5.14}$$

Like DADT-HD, the performance of this variant may suffer due to its use of document topics. However, it may outperform both DADT-HD and DADT-HA in cases where some authorship indicators are captured by the document topics, despite our expectation that document topics should mostly represent documents rather than authors.

Probabilistic: DADT-P. This method assumes that the test text was written by a previously-unknown author, infers the test text's document topic distribution $\tilde{\boldsymbol{\theta}}^{(D)}$ and the author/document topic ratio $\tilde{\pi}$, and returns the most probable author according to the following equation.

$$\underset{a\in\{1,\ldots,A\}}{\arg\max}\, p\left(\tilde{a}=a|\tilde{\boldsymbol{w}},\tilde{\pi},\tilde{\boldsymbol{\theta}}^{(D)},\boldsymbol{\theta}_a^{(A)},\boldsymbol{\Phi}^{(D)},\boldsymbol{\Phi}^{(A)},\chi_a\right)\propto \tag{5.15}$$

$$\underset{a\in\{1,\ldots,A\}}{\arg\max}\,\chi_a\prod_{i=1}^{\tilde{N}}\left(\tilde{\pi}\sum_{t=1}^{T^{(A)}}\theta_{at}^{(A)}\phi_{t\tilde{w}_i}^{(A)}+(1-\tilde{\pi})\sum_{t=1}^{T^{(D)}}\tilde{\theta}_t^{(D)}\phi_{t\tilde{w}_i}^{(D)}\right)$$

It is worth noting that in preliminary experiments, we found that an alternative approach that avoids sampling $\tilde{\pi}$ and $\tilde{\boldsymbol{\theta}}^{(D)}$ by setting $\tilde{\pi}=1$ yields poor performance, probably because it "forces" all the words to be author words, including words that are very likely to be document words. In addition, as discussed in Section 5.1.4, we found that following an approach where $\tilde{\pi}$ and $\tilde{\boldsymbol{\theta}}^{(D)}$ are sampled separately for each author (similarly to AT-FA-P2) yields comparable performance to sampling only once by following the previously-unknown author assumption. However, the former approach is too computationally expensive to run on datasets with many candidate authors. Hence, we present only the results obtained with the approach that performs sampling only once, as outlined above.

5.3 Evaluation

This section presents the results of our evaluation. We first describe our experimental setup (Section 5.3.1), followed by the results of our experiments on the Judgement and PAN'11 datasets (Sections 5.3.2 and 5.3.3 respectively). Then, we present the

results of a more restricted set of experiments on the larger IMDb62, IMDb1M and Blog datasets (Section 5.3.4).

5.3.1 Experimental Setup

We ran experiments on five datasets: Judgement, PAN'11, IMDb62, IMDb1M and Blog (Section 3.3). For all the datasets except PAN'11 we employed ten-fold cross validation, repeated with five different random seeds in the Judgement, IMDb62 and IMDb1M experiments (Section 3.1), and with a single random seed in the full Blog experiments (due to its large size and our resource constraints). PAN'11 experiments followed the setup of the PAN'11 competition (Argamon and Juola, 2011): we trained all the methods on the given training dataset, tuned the parameters according to results obtained for the given validation dataset, and ran the tuned methods on the given testing dataset. In all cases, we report the overall classification accuracy, i.e., the percentage of test texts correctly attributed to their author (Section 3.2).[3]

We used collapsed Gibbs sampling to train all the topic models (Sections 4.2.2 and 4.3.2), running 4 chains with a burn-in of 1,000 iterations. In the Judgement, PAN'11 and IMDb62 experiments, we retained 8 samples per chain with a spacing of 100 iterations. In the IMDb1M and Blog experiments, we retained 1 sample per chain due to runtime constraints. Since we cannot average topic distribution estimates obtained from training samples due to topic exchangeability (Steyvers and Griffiths, 2007), we averaged the distances and probabilities calculated from the retained samples. In the dimensionality reduction experiments, we used the topic distributions from a single training sample to ensure that the number of features is substantially reduced.[4] For test text sampling, we used a burn-in of 10 iterations and averaged the parameter estimates over the next 10 iterations in a similar manner to the procedure employed by Rosen-Zvi et al. (2010). We found that these settings yield stable results across different random seed values.

To enable a fair comparison between the topic-based methods and the Token SVM baseline, all the methods are trained on the same token representations of the texts. In most experiments, we do not apply any filters and simply use all the tokens as they appear in the text. In some cases, as indicated throughout this section, we either retain only stopwords or discard the stopwords in a preprocessing step that is applied before running the methods (the stopword list appears in Appendix A). This allows us to obtain rough estimates of the effect of considering

[3] In the PAN'11 case, statistical significance is reported according to the t-test on the sample level rather than on the fold level, because there is only one test fold (the validation dataset or the testing dataset).

[4] An alternative approach would be to use all the concatenation of all the samples, but this may result in a large number of features (e.g., $400 \times 8 \times 4 = 12,800$ when the number of topics is set to 400 and 8 samples are retained from 4 chains). In addition, we found in preliminary experiments that using this alternative approach did not improve results.

only style words, considering only content words and considering both style and content. However, we note that this is only a crude way of separating style and content, since some stopwords may contain content clues, while some words that do not appear in the stopword list may be seen as indicators of personal style, regardless of content (Section 2.3.2). Further, since our goal in subsequent chapters is to obtain representations of users' characteristics and interests, we use all the tokens in Chapters 6 and 7.[5]

As discussed in Section 4.1.2, in general, varying the number of topics has a larger effect on performance than varying the values of other configurable parameters. Hence, we present results obtained with different topic numbers, and set the priors to their default values for all the topic models, as specified in Section 4.1.2. The only exceptions are the $\delta^{(D)}, \delta^{(A)}$ and ϵ parameters of DADT, which we tuned in our Judgement and PAN'11 experiments, and set to the values that yielded the best performance for subsequent experiments (these values are: $\delta^{(D)} = 1.222, \delta^{(A)} = 4.889$ and $\epsilon = 0.009$).

5.3.2 Three-way Attribution of Judgements

The Judgement dataset is an example of a traditional authorship attribution dataset, as it contains relatively long texts written in a formal language, where all the texts share common themes (Section 3.3.3). We ran three-way authorship attribution experiments on this dataset, where the candidate authors are the three judges: Dixon, McTiernan and Rich. We first present the results obtained with LDA-based methods, followed by the results obtained with the AT model (with and without fictitious authors), and with our DADT-based methods, which yielded the best performance. We end this section with experiments that explore the effect of applying stopword filters to the corpus in a preprocessing step, which demonstrate that our DADT-based approach models authorship indicators other than content words.

LDA

Figure 5.1 presents the results of the LDA experiment. As the figure shows, the best performance obtained by training an SVM classifier on LDA topic distributions (LDA-SVM with 100 topics) was somewhat worse than that obtained by training directly on tokens (Token SVM), but was still much better than a majority baseline.[6] This indicates that although some authorship indicators are lost when using LDA for dimensionality reduction, many are retained despite the fact

[5]This is also the reason why we did not experiment with more sophisticated feature selection methods, such as information gain with regards to authors – such methods may result in token representations that are highly discriminative of authors, and thus do not align with our goal of obtaining a soft clustering of users (Chapter 4).

[6]The differences between LDA-SVM and the Token SVM and majority baselines are statistically significant in all cases.

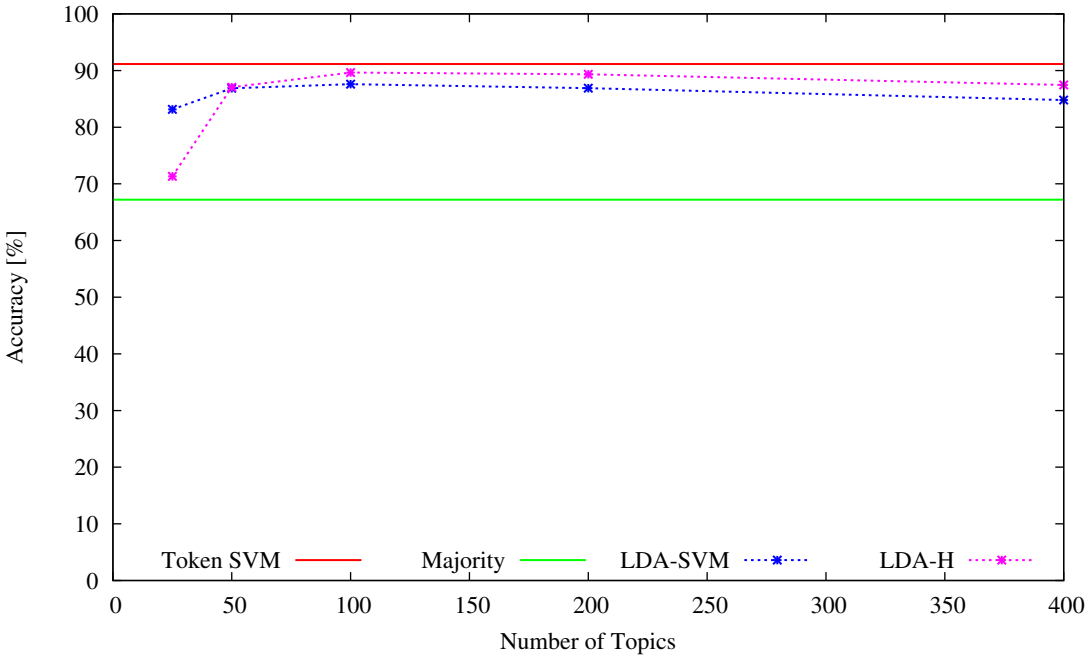


Figure 5.1: LDA results (dataset: Judgement)

that LDA's document representations are much more compact than the raw token representations.

Notably, the distance-based LDA-H outperformed the LDA-SVM approach when a sufficient number of topics (at least 100) was used, but LDA-H was still outperformed by the Token SVM baseline.[7] The reason for LDA-H outperforming LDA-SVM may be that LDA-SVM views each document as a separate instance, while LDA-H employs the mean distance of the test text's topic distribution from each author's set of training document distributions. This may have a de-noising effect, as the influence of each training document in LDA-H is smaller than in LDA-SVM, which means that, e.g., very short documents may be effectively ignored. This indicates that employing LDA to obtain author representations (as sets of document topic distributions) may have some merit, though this approach does not appear to capture all the relevant authorship indicators since LDA-H was outperformed by Token SVM.

AT

Figure 5.2 presents the results of the AT experiment. In contrast to LDA-SVM, using AT for dimensionality reduction (AT-SVM) yielded poor results. This is probably because the reduction is somewhat radical: each document is reduced to the same distribution over author topics because AT does not model individual documents (Section 5.2.3). Interestingly, AT-SVM's performance was very poor when 200

[7] The differences between LDA-H and LDA-SVM are statistically significant for all topic numbers except for 50. The differences between LDA-H and the Token SVM and majority baselines are statistically significant in all cases.

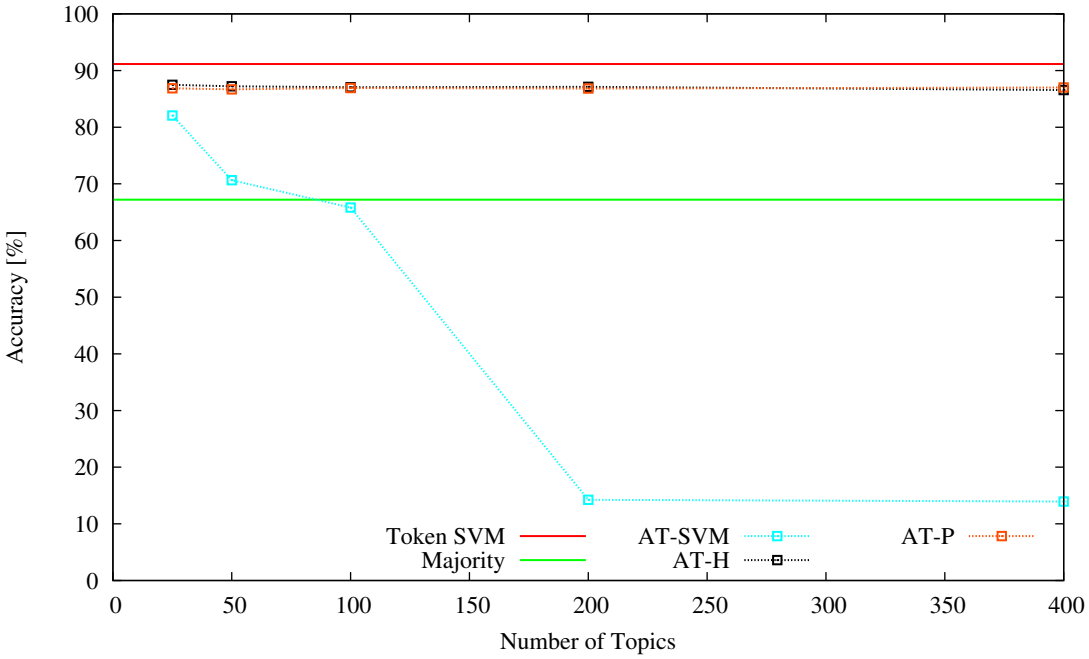


Figure 5.2: AT results (dataset: Judgement)

and 400 topics were used, possibly because the more fine-grained topic distributions yielded by employing more topics resulted in sparser author representations (where some topics were allocated only a few words), which made it hard for the SVM component to discriminate between texts by different authors.

Both the distance-based AT-H and the probabilistic AT-P significantly outperformed AT-SVM. Notably, only a small number of topics was required for either method to obtain its best performance. This is probably because the Judgement dataset contains only three authors, so only a few author topics are required to obtain topic distributions that vary sufficiently across authors. This may also be the reason why AT-H yielded performance that is slightly better than or comparable to AT-P's performance, despite our hypothesis that AT-P would outperform AT-H.[8] It appears that in this case, the author topic distributions are different enough for a distance measure to perform as well as a probabilistic measure.

When comparing AT-H and AT-P to the baselines, we see that both were outperformed by Token SVM, but yielded much better performance than the majority baseline.[9] We find these results encouraging, as they indicate that using AT topics to model authors captures many of the indicators required for authorship attribution, even though AT was not designed with authorship attribution in mind, and despite the fact that AT represents each author with a single distribution over topics while ignoring differences and similarities between documents. This stands in contrast to

[8]The differences between AT-H and AT-P are *not* statistically significant for all topic numbers except for 25.

[9]The differences between either AT-H or AT-P and the Token SVM and majority baselines are statistically significant in all cases.

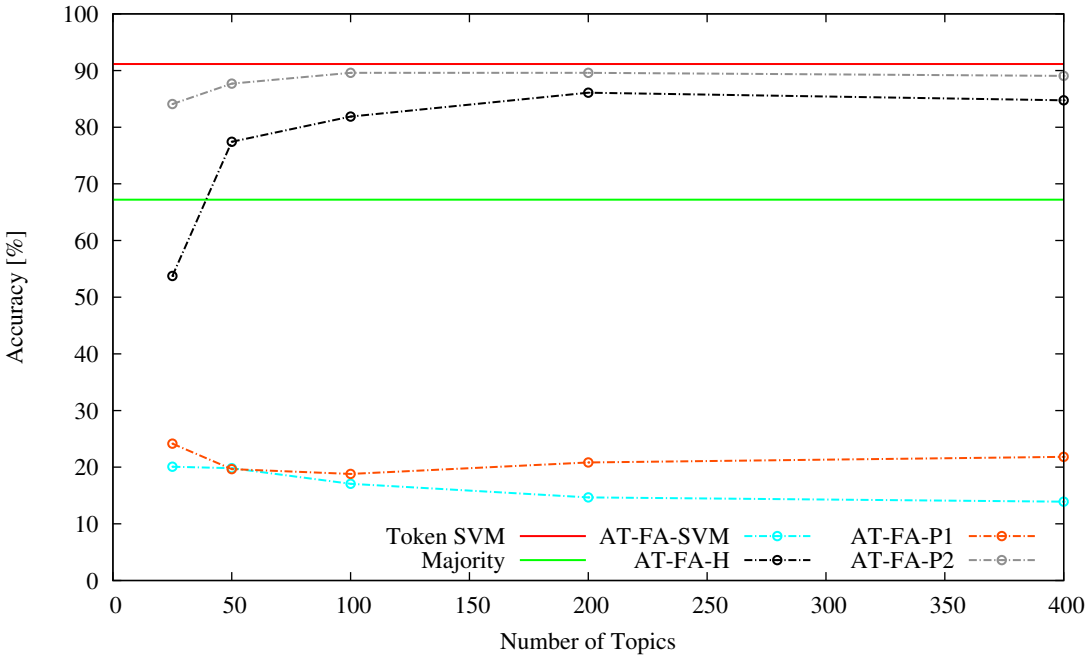


Figure 5.3: AT-FA results (dataset: Judgement)

Token SVM, which attempts to build a document-based model that is optimised for the classification goal of authorship attribution (Section 5.2.1).

AT-FA

Figure 5.3 presents the results of the AT-FA experiment. As we can see, in this case the AT-FA-SVM approach performed more poorly than the corresponding method in the AT case *without* fictitious authors (AT-SVM in Figure 5.2): AT-FA-SVM's highest accuracy of 20.09% (obtained with 25 topics) is significantly worse than AT-SVM's best accuracy of 82.07% (also obtained with 25 topics). This may seem surprising, since the only difference between AT and AT-FA is the addition of a fictitious author for each document, which was shown to improve AT's ability to predict unseen portions of documents (Rosen-Zvi et al., 2010). However, the reason for AT-FA-SVM's poor performance may be that it does not allow us to conserve the underlying assumption of fictitious authors in the classification stage – we cannot assume that the test text was written by a fictitious author together with a previously-unseen author (Section 5.2.4). This is probably also the reason why the probabilistic AT-FA-P2 significantly outperformed AT-FA-P1 by a large margin in all cases – AT-FA-P2 conserves the fictitious author assumption, while AT-FA-P1 ignores it.

Somewhat surprisingly, AT-FA-H significantly outperformed both AT-FA-SVM and AT-FA-P1 in all cases, even though it also violates the underlying assumption of fictitious authors in the classification stage. This may be because AT-FA-H is less

Method	**Accuracy**
Majority	67.21%
Token SVM	91.15%
LDA-H	89.65%
DADT-SVM	85.49%
DADT-HD	38.27%
DADT-HA	90.50%
DADT-HDA	91.10%
DADT-P	**93.64%**

Table 5.1: DADT results (dataset: Judgement)

sensitive to the violation in the model assumptions than either AT-FA-SVM or AT-FA-P1: (1) unlike AT-FA-SVM, which tries to resolve the issue by duplicating the test text's inferred topic distribution, AT-FA-H simply ignores the violation in the model assumptions; and (2) unlike AT-FA-P1, AT-FA-H is not directly reliant on the model's probabilistic structure. As in the AT-H case, it appears that considering distance yields relatively good results, at least on the Judgement dataset. However, AT-FA-H was significantly outperformed by AT-FA-P2 in all cases. This is in line with our expectations, as AT-FA-P2 is the only method that conserves the fictitious author assumption in the classification phase.

When comparing AT-FA-P2 to the baselines, we see that it was outperformed by Token SVM for all topic numbers, but yielded better performance than the majority baseline.[10] Despite the fact that AT-FA-P2 was outperformed by Token SVM, the margin was not large when enough topics were used (AT-FA-P2 yielded its best accuracy of 89.60% with 100 topics, in comparison to Token SVM's accuracy of 91.15%). This indicates that representing both documents and authors in the topic model may have advantages in terms of authorship attribution. This further motivates the use of our DADT model, which considers documents and authors without requiring the preprocessing step of adding fictitious authors.

DADT

Table 5.1 presents the results of the DADT experiment, obtained with 10 author topics, 90 document topics, and prior settings of $\delta^{(D)} = 1.222$, $\delta^{(A)} = 4.889$ and $\epsilon = 0.009$ (other parameter settings are discussed below). These results are compared to the baselines (majority and Token SVM), and to the best topic-based result obtained thus far (by LDA-H with 100 topics). As we can see, the best DADT-based result (highlighted in boldface) was obtained with the probabilistic DADT-P

[10]The differences between AT-FA-P2 and the Token SVM and majority baselines are statistically significant in all cases.

method, which significantly outperformed all the other methods. This demonstrates the effectiveness of our DADT model in capturing author characteristics that are relevant to authorship attribution, thereby providing further evidence to support our topical user modelling approach.

Notably, DADT-SVM and DADT-HD yielded significantly poorer results than DADT-HA, DADT-HDA and DADT-P. DADT-SVM's performance may be because its use of document topics introduces noise that causes the SVM component to underperform, as DADT's document topics are not expected to be indicative of authorship. Similarly, we expected DADT-HD to perform poorly since it relies only on distance between document topics – the fact that it yielded poor performance serves as evidence that DADT works as expected and separates document and author words into document and author topics respectively. However, combining document topic distance with author topic distance as done by the DADT-HDA method significantly improved performance by a small margin over the DADT-HA method (which relies only on author topic distance), possibly because some authorship indicators were still captured by document topics. The reason why DADT-HDA was more effective in combining author and document topics than DADT-SVM may be similar to reason why LDA-H outperformed LDA-SVM (Figure 5.1): like LDA-H, DADT-HDA employs the mean distance to each author's documents, rather than considering each document separately (as done by LDA-SVM and DADT-SVM). As mentioned above in our comparison of LDA-H and LDA-SVM, this appears to have a de-noising effect, at least on the Judgement dataset.

Our choice of DADT settings reflects the following insights:

- We used 100 topics overall based on the results of the other topic-based methods, which showed that good results are obtained with this number of overall topics. We chose the 90/10 document/author topic split because in this case DADT attempts to model only three authors who wrote many documents.
- Setting $\delta^{(D)} = 1.222$ and $\delta^{(A)} = 4.889$ encodes our prior belief that the portion of each document that is composed of author words is 80% on average, with 15% standard deviation ($\delta^{(D)}$ and $\delta^{(A)}$ were obtained according to DADT's definition, as explained in Section 4.1.2).
- Setting $\epsilon = 0.009$ encodes our prior belief that stopword choice is more likely to be influenced by the identity of the author than by the content of the documents (Section 4.1.2).

Somewhat surprisingly, these settings did not have a large effect on the performance of the methods in most cases. This is demonstrated by the results presented in Table 5.2, which were obtained by varying the values of the above parameters and running the DADT-P method. As the table shows, the results obtained with a

$T^{(D)}$	$T^{(A)}$	$\delta^{(D)}$	$\delta^{(A)}$	ϵ	**Accuracy**
90	10	1	1	0	**93.81%**
90	10	1.222	4.889	0	**93.49%**
90	10	1.222	4.889	0.009	**93.64%**
50	50	1.222	4.889	0.009	92.88%
10	90	1.222	4.889	0.009	88.62%

Table 5.2: DADT-P tuning results (dataset: Judgement)

setting of $\delta^{(D)} = \delta^{(A)} = 1$, which can be seen as encoding no strong prior belief about the document/author word balance in each document (it is equivalent to setting a uniform prior on this balance), were comparable to the results obtained with $\delta^{(D)} = 1.222$ and $\delta^{(A)} = 4.889$. Likewise, changing ϵ from 0 to 0.009 only had a minor effect on the results. The only setting that made a relatively large difference is the document/author topic split – changing it from 90/10 to 10/90 yielded poorer results. However, the 50/50 split yielded close results to the 90/10 split, which shows that in this case, the document/author topic split setting is sensitive to relatively large variations.

It is likely that performing an exhaustive grid search for the optimal parameter settings for each method would allow us to obtain somewhat improved results. However, such a search would be computationally expensive, as the model needs to be retrained and tested for each fold, parameter set and method. Therefore, we decided to present the results obtained with the non-optimised settings, which are good enough to demonstrate the merits of our DADT approach, as DADT-P outperformed all the other methods discussed so far.

Testing the Effect of Stopwords

The results reported up to this point were all obtained by running the methods on document representations that include all the tokens. As discussed in Section 5.3.1, discarding or retaining stopwords provides a crude way of separating style from content. We ran a set of experiments where we either discarded stopwords in a preprocessing step or retained only stopwords, and then ran the Token SVM baseline and the DADT-P method, which obtained the best performance when all the tokens were used (DADT was run with the same settings used to obtain the results presented in Table 5.1).

The results of this experiment are presented in Table 5.3. As the results show, discarding stopwords caused the Token SVM baseline to yield poorer performance than when all the tokens were used, but retaining only stopwords significantly improved Token SVM's performance. Interestingly, this was not the case with DADT-P, where either discarding or retaining stopwords caused a statistically significant

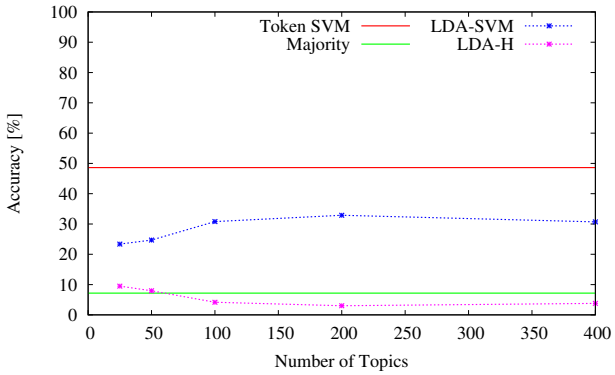
Method	**Accuracy**
Majority	67.21%
Token SVM	
All tokens	91.15%
No stopwords	86.18%
Only stopwords	92.76%
DADT-P	
All tokens	**93.64%**
No stopwords	89.28%
Only stopwords	90.85%

Table 5.3: Stopword experiment results (dataset: Judgement)

drop in performance in comparison to using all the tokens. The reason why DADT-P's performance dropped when only stopwords were used may be that DADT was designed under the assumption that all the tokens in the corpus are retained. Hence, DADT may underperform when this assumption is violated. However, we are encouraged by the fact that DADT-P's performance drop was not very large when only stopwords were retained (it still significantly outperformed the majority baseline by a large margin and obtained comparable performance to Token SVM when trained on all the tokens), as it indicates that DADT captures stylistic elements in the authors' texts.

Another encouraging result is that DADT-P yielded significantly better performance than Token SVM when using feature sets that included all the tokens or all the tokens without stopwords. DADT-P appears to harness the extra information from non-stopword tokens more effectively than Token SVM, despite the fact that such tokens tend to occur less frequently in the texts than stopwords. Further, the vocabulary size of these two feature sets is larger than that of the stopword-only feature set, which suggests that DADT-P is more resilient to noise than Token SVM.

It is worth noting that some content-independent data is lost when only stopwords are retained. For example, the phrase "in my opinion" appears in texts by all three authors, but it is used more frequently by McTiernan (it occurs in about 82% of his judgements) than by Dixon (69%) or Rich (58%). As the frequency of this phrase is apparently dependent on author style and independent of the specific content of a given judgement, it is probably safe to assume that it would be beneficial to retain the word "opinion". However, this word does not appear in our stopword list. A possible solution is to obtain corpus-specific stopwords, e.g., by extracting a list of frequent words, but this gives rise to new problems, such as determining a frequency threshold. We decided not to pursue such a solution, as our main goal is



(a) Validation

Method	$T^{(D)}$	Accuracy
Majority	—	7.15%
Token SVM	—	**53.31%**
LDA-SVM	200	31.92%
LDA-H	25	10.62%

(b) Testing

Figure 5.4: LDA results (dataset: PAN'11)

to obtain compact representations of users that capture both their authorship style and their interests.[11] Further, even though Token SVM performed better when only stopwords were retained, this behaviour is not stable across datasets, as will be shown in the next section.

5.3.3 Email Attribution with Tens of Authors

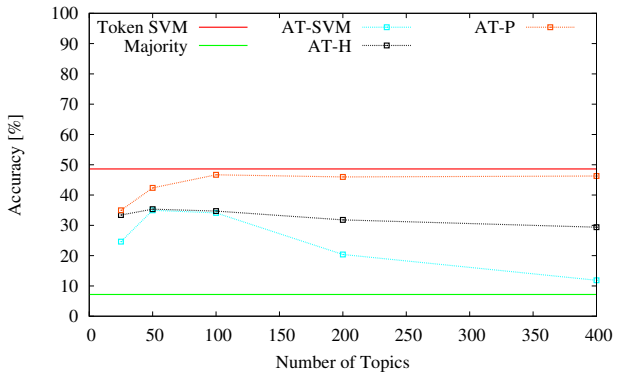
In this section, we present the results of our experiments on the PAN'11 dataset, which contains emails by 72 different authors (Section 3.3.4). Authorship attribution on the PAN'11 dataset is more challenging than on the Judgement dataset, since PAN'11 texts are shorter and more informal than judgements, and some of the PAN'11 authors wrote only a few emails. This section is organised in a similar manner to the previous section, with LDA, AT, AT-FA and DADT experiments appearing in this order, followed by experiments that explore the effect of stopword filters. In all cases, we tune the methods on the validation subset and report the results obtained on the testing subset with the settings that yielded the best validation results (i.e., each method is run multiple times on the validation subset and a single time on the testing subset).[12]

LDA

Figure 5.4 presents the results of the LDA experiment. These results stand in sharp contrast to the results obtained on the Judgement dataset. On PAN'11, LDA-H performed much more poorly than LDA-SVM, which was clearly outperformed by Token SVM (on Judgement the differences between the three methods were much

[11] It is worth noting that we did not expect to obtain interest representations from the texts in the Judgement dataset, because all three judges handled cases of all types. However, interests may vary from author to author in the other datasets we consider. For example, IMDb users may choose to write only about movies from certain genres (but it is unlikely that relying only on genre-specific words would be enough to obtain good authorship attribution results, since all genres are expected to interest at least several users).

[12] For most methods, testing results are better than the best validation results. This may be because on average testing texts are about 10% longer than validation texts (Section 3.3.4).



(a) Validation

Method	$T^{(A)}$	Accuracy
Majority	—	7.15%
Token SVM	—	**53.31%**
AT-SVM	50	39.23%
AT-H	50	40.62%
AT-P	100	**53.08%**

(b) Testing

Figure 5.5: AT results (dataset: PAN'11)

smaller). The reason for this difference may be that LDA does not consider authors in the model-building stage. While this had a relatively small effect on performance in the three-way Judgement attribution scenarios, it appears that accounting for authors is important in scenarios with many authors. As the rest of this thesis deals with such scenarios, we decided not to use LDA for modelling authors in subsequent sections and chapters.

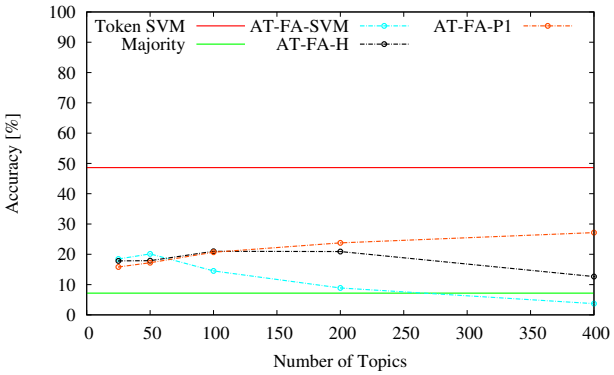
AT

Figure 5.5 presents the results of the AT experiment. As in the LDA experiments, the transition from the Judgement dataset to the PAN'11 dataset with its 72 authors allows us to get a clearer view of how the methods perform in comparison to each other. Specifically, here we see that, as we expected, the probabilistic AT-P method yielded the best results. This stands in contrast to the Judgement results, where AT-P and AT-H performed similarly. Further, AT-P yielded comparable performance to that of Token SVM, and obtained higher accuracy than that obtained by the LDA-based methods (Figure 5.4). This supports our claim that authors should be considered by the underlying topic model if we wish to obtain good author representations.

AT-FA

Figure 5.6 presents the results of the AT-FA experiment. When comparing the AT-FA results to the AT results (Figure 5.5), we can see that in this case adding fictitious authors consistently yielded poorer performance. This may be because the methods we tested do not satisfy the fictitious author assumption in the classification phase (Section 5.2.4).[13] As we will show below, this issue is addressed by our DADT model, which is aware of both authors and documents, and does not require the addition of fictitious authors.

[13]We did not run the AT-FA-P2 method because it requires running a separate sampling chain for each candidate author and test text (Section 5.2.4). Hence, it is too computationally expensive to run in cases with many candidate authors and test texts.



(a) Validation

Method	$T^{(A)}$	**Accuracy**
Majority	—	7.15%
Token SVM	—	**53.31%**
AT-FA-SVM	50	23.15%
AT-FA-H	100	24.92%
AT-FA-P1	400	29.69%

(b) Testing

Figure 5.6: AT-FA results (dataset: PAN'11)

$T^{(D)}$	$T^{(A)}$	$\delta^{(D)}$	$\delta^{(A)}$	ϵ	**Accuracy**
10	90	1	1	0	48.53%
10	90	1.222	4.889	0	53.40%
10	90	1.222	4.889	0.009	**54.86%**
50	50	1.222	4.889	0.009	50.31%
90	10	1.222	4.889	0.009	30.48%

Table 5.4: DADT-P tuning results (dataset: PAN'11 Validation)

DADT

We ran the DADT experiments with 100 topics overall, as this number of topics yielded the best topic-based results of the models and methods whose results we presented thus far (as shown by a comparison of Figure 5.5 to Figures 5.4 and 5.6, AT-P with 100 topics yielded the best results of the methods based on LDA, AT and AT-FA). Table 5.4 shows the results of tuning DADT's settings and running DADT-P on the PAN'11 validation set. As with the other results presented in this section, the PAN'11 tuning experiment shows a clearer picture in terms of accuracy differences between different parameter settings than in the Judgement experiments. Specifically, when we used uninformed uniform priors on the document/author word split ($\delta^{(D)} = \delta^{(A)} = 1$), and the same word-in-topic priors for both document and author words ($\epsilon = 0$), the obtained accuracy was comparable to AT-P's accuracy. On the other hand, setting $\delta^{(D)} = 1.222$ and $\delta^{(A)} = 4.889$, which encodes our prior belief that on average 80% (with a standard deviation of 15%) of each document is composed of author words, significantly improved performance. Setting $\epsilon = 0.009$ to encode our prior knowledge that stopwords are indicators of authorship yielded an additional improvement. Finally, the last two results in Table 5.4 demonstrate the importance of having enough topics to model the authors – accuracy dropped by about 4 percentage points when we used 50 author topics and 50 document topics,

Method	Validation	Testing
Majority	7.18%	7.15%
Token SVM	48.61%	53.31%
DADT-SVM	34.95%	39.69%
DADT-HD	0.39%	0.15%
DADT-HA	31.10%	38.23%
DADT-HDA	20.60%	25.23%
DADT-P	**54.86%**	**59.38%**

Table 5.5: DADT results (dataset: PAN'11)

and by about 24 percentage points when we used only 10 author topics and 90 document topics, rather than 90 author topics and 10 document topics. This leads us to conjecture that it would be beneficial to pursue a future extension that learns the topic balance automatically, e.g., in a similar manner to Teh et al.'s (2006) method of inferring the number of topics in LDA.

Table 5.5 presents the results obtained with all the DADT-based methods, using the best setting from Table 5.4: 10 document topics, 90 author topics, $\delta^{(D)} = 1.222$, $\delta^{(A)} = 4.889$ and $\epsilon = 0.009$. As the table shows, DADT-P, which obtained the best performance of all the methods tested in this section, is the only method that outperformed Token SVM. This implies that our DADT model is the most suitable of the models we considered for capturing patterns in the data that are important for authorship attribution, at least in scenarios that are similar to the scenario represented by the PAN'11 dataset.

The properties of DADT are also demonstrated by the difference between DADT-HA and DADT-HD: DADT-HA, which is based on *author* topic distance, outperformed DADT-HD, which is based on *document* topic distance, meaning that our DADT model worked as expected and separated *author* words from *document* words. Interestingly, in contrast to Judgement, in this case combining the document-based distance with the author-based distance (DADT-HDA) yielded worse performance than DADT-HA. This is probably because the DADT-HD results were much weaker on PAN'11 than on Judgement, and DADT-HDA linearly combines the distances used by DADT-HD and DADT-HA by giving them equal weights (Equation 5.14). By contrast, DADT-SVM also combines the document topic distributions with the author topic distributions, but learns how well they help discriminate texts by different authors, thereby outperforming both DADT-HDA and DADT-HA.

DADT-P's testing result is comparable to the third-best accuracy (out of 17) obtained in the PAN'11 competition (Argamon and Juola, 2011).[14] However, to the

[14]Competitors were ranked according to macro-averaged and micro-averaged precision, recall and F1. In this case, the micro-averaged measures are all equivalent to the accuracy measure because each of the test texts is assigned to a single candidate author.

Method	Validation	Testing
Majority	7.18%	7.15%
Token SVM		
All tokens	48.61%	53.31%
No stopwords	42.36%	46.46%
Only stopwords	27.78%	28.38%
DADT-P		
All tokens	**54.86%**	**59.38%**
No stopwords	50.62%	54.69%
Only stopwords	18.06%	18.54%

Table 5.6: Stopword experiment results (dataset: PAN'11)

best of our knowledge, DADT-P obtained the best accuracy for a fully-supervised method that uses only unigram features. Specifically, Kourtis and Stamatatos (2011), who obtained the highest accuracy (65.8%), assumed that all the test texts are given to the classifier at the same time and used this additional information with a semi-supervised method, while Kern et al. (2011) and Tanguy et al. (2011), who obtained the second-best (64.2%) and third-best (59.4%) accuracies respectively, used various feature types (e.g., features obtained from parse trees). In addition, preprocessing differences make it hard to compare the methods on a level playing field. Nonetheless, we note that extending DADT to enable semi-supervised classification and additional feature types are promising directions for future work (Section 8.2).

Testing the Effect of Stopwords

Table 5.6 shows the results of an experiment where we applied stopword filters to the corpus and ran the Token SVM and DADT-P methods. In this case, retaining only stopwords significantly hurt the performance of both methods, which stands in contrast to the Judgement stopword experiment, where retaining stopwords improved the performance of the Token SVM baseline. The reason for this may be that other words beyond stopwords are also indicative of authorship in this dataset. For instance, Tanguy et al. (2011) used the openings and closings of the emails in the dataset as separately-weighted features. Openings can start with words such as "hello", "hi", "hey", and "dear", but only the first two words appear in our stopword list, meaning that even when stopwords are retained some stylistic features are lost. This, again, highlights the difficulties in extracting words that are truly content-independent – a problem that would be especially relevant when trying to adapt an authorship classifier trained on texts from one domain to texts from a completely different domain. However, such problems are beyond the scope of this

Method	IMDb62	IMDb1M	Blog (prolific)	Blog (full)
Majority	7.37%	3.00%	1.28%	0.62%
Token SVM	**92.52%**	43.85%	32.96%	24.13%
AT-P	89.62%	40.82%	37.59%	23.03%
DADT-P	91.79%	**44.23%**	**43.65%**	**27.63%**

Table 5.7: Large-scale experiment results (datasets: IMDb62, IMDb1M and Blog)

study. Hence, in the remainder of this thesis we do not apply any stopword filters to the corpora we use.

5.3.4 Experiments on Large Datasets

In this section, we report the results of our experiments on the IMDb62, IMDb1M and Blog datasets (Section 3.3). Both IMDb datasets contain movie reviews and message board posts, with IMDb62 consisting of texts by 62 prolific authors (with at least 1,000 texts each), and IMDb1M consisting of texts by 22,116 authors, who are mostly non-prolific. The Blog dataset contains blog posts by 19,320 authors, and is the largest of the datasets we considered in terms of token count – it contains about 168 million tokens, while IMDb62 and IMDb1M contain about 22 and 34 million tokens respectively. In addition to running experiments on the full Blog dataset, we considered a subset that contains all the texts by the 1,000 most prolific authors (this subset contains about 69 million tokens overall in 332,797 posts – about 49% of the posts in the full Blog dataset).

Due to resource constraints, we performed a more restricted set of experiments on IMDb62, IMDb1M and Blog than on the Judgement and PAN'11 datasets (which contain about 3 and 0.74 million tokens respectively). We ran only the Token SVM baseline, AT-P and DADT-P, as these methods yielded the best performance in the PAN'11 experiments. We set the overall number of topics of AT and DADT to 200 topics for IMDb62, and 400 topics for IMDb1M and Blog. We set DADT's document/author topic split to 50/150 for IMDb62, 50/350 for IMDb1M, 50/350 for Blog (prolific) and 10/390 for Blog (full), and used the prior setting that yielded the best PAN'11 results ($\delta^{(D)} = 1.222, \delta^{(A)} = 4.889$ and $\epsilon = 0.009$).[15]

Table 5.7 shows the results of this set of experiments. As in our previous experiments, DADT-P consistently outperformed AT-P, which indicates that employing disjoint sets of document and author topics yields author representations that are more suitable for authorship attribution than employing only author topics. In contrast to the previous experiments, Token SVM outperformed DADT-P in one case,

[15] As in the PAN'11 experiments, we determined the overall number of topics based on AT-P's performance with 25, 50, 100, 200 and 400 topics. The document/author topic splits we tested were 10/190, 50/150 and 100/100 for IMDb62, and 10/390, 50/350 and 100/300 for IMDb1M and Blog.

the IMDb62 dataset. This may be because discriminative methods (such as Token SVM) tend to outperform generative methods (such as DADT-P) in scenarios where training data is abundant (Ng and Jordan, 2001), which is the case with IMDb62 that contains at least 900 texts per author in each training fold.

A notable result is that while all the methods yielded relatively low accuracies on the full Blog dataset, the topic-based methods experienced a larger drop in accuracy than Token SVM when transitioning from the prolific author subset to the full dataset. This may be because topic-based methods employ a single model, making them more sensitive to the number of authors than Token SVM's one-versus-all setup that uses one model per author (this sensitivity may also explain why DADT-P outperformed Token SVM by a relatively small margin on IMDb1M). This result suggests a direction for future work in the form of an ensemble of Token SVM and DADT-P. The potential of this direction is demonstrated by the fact that a perfect oracle, which chooses the correct answer between Token SVM and DADT-P when they disagree, yields an accuracy of 37.15% on the full Blog dataset.

5.4 Application to Reviewer Identification

AT and DADT can potentially be used to identify anonymous reviewers based on publicly-available data – the reviewer list (which is commonly available), and their published papers. The main question in this case is whether authorship markers learned from (often multi-authored) texts in one domain (the papers) can be used to classify single-authored texts from a related domain (the reviews).

To start answering this question, we considered a small conference track, which attracted 18 submissions that were each reviewed by two reviewers. We collected the bodies of 10 papers (i.e., without references, author names, acknowledgements, etc.) by each of the 18 reviewers that were listed in the proceedings, which resulted in a training corpus of 171 documents with 196 authors overall (some of the reviewers have co-authored papers). We omitted authors with only one paper, since their presence is equivalent to having fictitious authors, which may hurt performance (Section 5.3.3). This resulted in a total of 77 authors. Our test dataset consisted of 19 reviews by the 9 reviewers that gave us permission to use their reviews.

We trained AT and DADT on the paper corpus under the setup described in Section 5.3.1, and used AT-P and DADT-P to classify the reviews. The best accuracy, 8/19, was obtained by DADT-P with 10 document topics and 90 author topics. The accuracy of AT-P (with 100 topics) was slightly worse at 7/19. In addition, the correct reviewer appeared in the top-five list of probable authors for 15/19 of the reviews with DADT-P and 11/19 with AT-P.[16]

[16]The list of probable authors included all 18 reviewers – we considered all the reviewers as candidates since this did not require using any private information and it made our experimental setup more realistic.

We obtained better results by completely eliminating non-reviewers from the training corpus. DADT-P required only 25 document topics and 25 author topics in this case, and its accuracy rose to 10/19 (AT-P again performed worse with an accuracy of 7/19). In 16/19 of the cases the correct reviewer appeared in DADT-P's top-five list, compared to 12/19 with AT-P.

Our results are preliminary, as they were obtained on a very small dataset. Still, they indicate that reviewer identification is feasible (note that it is unlikely that DADT-P's performance is only due to content words, as interest areas are often shared between reviewers). To verify this, a fully-fledged study should be done on a corpus of reviews from a large conference, with a training corpus that includes each author's full body of publications (perhaps dropping very old publications, which we did not do). As far as we know, such a study is yet to be performed. The closest work we know of is by Nanavati et al. (2011), who considered the question of whether "insiders", who served as program committee members and thus had access to non-anonymous reviews, can use these reviews as training data to identify reviewers. While they found that they could identify reviewers with high accuracy, the main limitation of their approach is that it relies on private data.

Nonetheless, we believe that reviewer anonymity needs to be seriously addressed. One approach is to use tools that obfuscate author identity, as developed by, e.g., Kacmarcik and Gamon (2006) and Brennan and Greenstadt (2009). However, as this may lead to an "arms race" between such tools and authorship analysis methods, perhaps the best approach is to forgo anonymity completely, as advocated by some researchers and editors (Groves, 2010). This is an open question with no simple answers, but we hope that our results will help motivate the search for solutions.

5.5 Summary and Conclusions

In this chapter, we introduced methods for authorship attribution that are based on the topical user models from Chapter 4, and tested them in several scenarios where the number of authors varies from three to about 20,000. We showed that in most cases, a probabilistic approach that is based on our DADT model (DADT-P) yielded the best results, outperforming methods based on the LDA and AT topic models, as well as a Token SVM baseline. This indicates that our topical user modelling approach successfully captures indicators of user style (which is indicative of user characteristics such as demographic attributes and personality traits) as reflected by their texts.

As one of the first studies on applying topic modelling techniques to authorship attribution, the work presented in this chapter can be extended in many ways. One direction is enabling the use of various feature types, e.g., by incorporating conditional random fields into DADT in a similar manner to Zhu and Xing's (2010)

model. This direction can also be pursued by using DADT-P in an ensemble with SVMs that can trained on feature types other than token unigrams – this may also have the added value of combining the strengths of DADT with those of the SVM approach (Section 5.3.4). Other potential extensions are employing the topic models in semi-supervised scenarios, and inferring the optimal number of author and document topics – both of these extensions require changes only to the training procedure rather than changes to the actual models.

In the rest of this thesis, we use the conclusion that DADT, and to a lesser extent AT (which is a private case of DADT) successfully represent authors based on their texts. We employ these two models to measure similarity between users based on their texts in our polarity inference framework (Chapter 6), and to obtain compact representations of users for our rating prediction framework (Chapter 7). Our results show that both models may be used to improve performance when employed within these two frameworks.

Chapter 6

User-aware Polarity Inference

In Chapter 5, we investigated aspects of user identity by applying topical user models to the authorship attribution task. In this chapter, we move on to explore the connection between user identity and sentiments expressed in user-generated texts by addressing the polarity inference task in a user-aware manner.

As discussed in Section 2.4.1, polarity inference is one of the key tasks in the sentiment analysis field. The binary case consists of classifying texts as either positive or negative. Less attention has been paid to the more challenging task of *multi-way* polarity inference, i.e., inferring the "star rating" of texts on a scale of more than two values.[1] In addition, little research has been done on harnessing the identity of authors to improve the accuracy of polarity inference methods (Section 2.4.2).

Two key challenges in polarity inference are:

- Different people often use *different* language to express the *same* thing, or conversely, use the *same* language to express *different* things (i.e., texts by different users may be discordant). Examples for the former are abundant, as people's choice of words often varies (e.g., "that's excellent" and "that's great" are both likely to be positive). An example for the latter comes from the humorous Anglo-EU Translation Guide,[2] where the British phrase "that's not bad" is intended to mean "that's good", but is understood by non-British English speakers to mean "that's poor".
- In multi-way polarity inference, polarities on a non-binary scale are more open to interpretation than binary polarities, meaning that every user has a different "feel" for the polarity scale. For example, the difference between a rating of

[1] It is important to note that while the word "rating" may be used to refer to non-binary polarity *values*, the polarity inference task is completely different from the rating prediction task, which we address in Chapter 7. While in both cases the output is a numeric rating, in polarity inference the input is a document and the output rating represents the level of sentiment polarity expressed in the document, and in rating prediction the input is a tuple of user and item identifiers and the output rating represents the user's predicted level of sentiment towards the item.

[2] Numerous copies of this guide exist, and thus we could not find its original source. See `languagelog.ldc.upenn.edu/nll/?p=3154` for a discussion of possible sources.

6 and a rating of 7 on a 10-point scale is less clear-cut than the difference between positive and negative.

We address these two challenges by considering users when performing polarity inference. Specifically, we introduce a framework that infers the polarities of texts written by a given *target user* based on user-specific polarity inference models, where the models can be weighted according to user similarity. We consider several approaches to combining the outputs of the users' models, and show that our best approach outperforms two popular baselines, even when all models are given equal weights. We also introduce several similarity measures that are based on different aspects of user language (e.g., as captured by topical user models) and rating patterns. Employing these similarity measures yields further improvements in the performance of our framework.

This chapter is organised as follows. Section 6.1 defines terms that are used throughout this chapter, and Section 6.2 describes baseline approaches to polarity inference. Sections 6.3 and 6.4 present our polarity inference framework and the user similarity measures it employs, respectively. Section 6.5 presents the results of our evaluation, and Section 6.6 concludes the chapter.

6.1 Definitions

We use the following terms throughout this chapter:

- *Labelled text*: A sentiment-bearing document with known polarity. We denote such a document with q to differentiate it from a document d, which is not known to bear sentiment. We assume that the user u who wrote the text is known, and denote the polarity rating of the text with r_{uq}.
- *Test text*: A sentiment-bearing document whose polarity we want to infer, and for which we know the author. We denote the polarity inferred by a given algorithm with $\hat{r}_{uq}$.
- *Target user*: The author of the test text. The target user may be a new user, for whom a few or no labelled texts are available.
- *Training users*: The authors of labelled texts, excluding the target user.
- *Inferrer*: A model that is trained on labelled texts and outputs the polarities of test texts. Examples for inferrers include generic classifiers (with discrete outputs), regression models (with continuous outputs), and models specific to polarity inference, such as those discussed in Section 2.4.1.

6.2 Baseline Approaches to Polarity Inference

We considered two baseline approaches to polarity inference. The first, *Single Inferrer, Multiple Users* (SIMU), simply ignores any information about the users who

wrote the texts (Section 6.2.1). The second, *Single Inferrer, Single User* (SISU), uses some knowledge about the users by employing user-specific inferrers (Section 6.2.2).

6.2.1 Single Inferrer, Multiple Users

The approach taken by the vast majority of researchers in the sentiment analysis field is SIMU (Section 2.4). This approach ignores any knowledge about the authors of the texts, and simply trains a single inferrer on the full set of labelled texts.[3] The fact that the SIMU approach is so widespread is unsurprising, since sentiment analysis originally arose as an area in natural language processing, where the focus is mainly on linguistic aspects of problems rather than on users.

There are two potential problems with the SIMU approach. First, SIMU may yield poor performance due to discordances between labelled texts by many different users. This is demonstrated by the examples given at the beginning of this chapter, which showed that some phrases are positive to some users and negative to others, and that users have different interpretations of non-binary polarity scales. Second, when many labelled texts are available, training a SIMU model with some inference algorithms may become too computationally expensive. To address these problems, one could randomly sample a subset of the labelled texts and use this subset for training, but this may not result in satisfactory performance (Section 6.5.2).

6.2.2 Single Inferrer, Single User

The SISU approach was introduced by Pang and Lee (2005), mostly to address the differences between users' interpretations of the polarity scale. SISU addresses SIMU's poor performance when labelled texts by many different users are available by training a separate inferrer for each user. SISU then uses only the target user's inferrer to infer the polarities of texts by the target user.

The main disadvantage of the SISU approach is that it may require many labelled texts by the target user to achieve acceptable performance. In addition, SISU cannot be used when there are no labelled texts by the target user. A possible solution in this case is to randomly select another user's inferrer, but this is unlikely to perform well. In our experiments, we verified the existence of these problems (Section 6.5.2). This led us to develop our approach, which is described in the following section.

6.3 Our User-aware Approach

Our approach, *Multiple Inferrers, Multiple Users* (MIMU), addresses SIMU's and SISU's shortcomings by training a separate inferrer for each user and combining the inferrers' outputs. Training a separate inferrer for each user addresses SIMU's potential difficulties with generalising from many possibly discordant texts and with training a single model on large amounts of data (Section 6.2.1). In addition, MIMU

[3]This inferrer may comprise multiple sub-inferrers, but as a whole it is unaware of the authors.

Variant	Equations	Requires Labelled Target User Texts?
MIMU-WNA	6.1	No
MIMU-NNI (mean deviation)	6.2	Yes
MIMU-NNI (z-score)	6.3	Yes
MIMU-TUI (mean deviation)	6.4	Yes
MIMU-TUI (z-score)	6.5	Yes
MIMU-TUI (static weight)	6.6 & 6.7	No
MIMU-TUI (dynamic weight)	6.6 & 6.8	Yes

Table 6.1: MIMU variants

addresses SISU's limitations by making it possible to learn from labelled texts by the target user, while enabling inference even when few target user texts are available (Section 6.2.2).

There are many possible ways of combining the polarities inferred by each user's inferrer. The remainder of this section introduces the combination variants we experimented with, and discusses the rationale behind each one of them. Table 6.1 summarises the names of the variants, the equations that define them, and whether or not they require labelled texts by the target user.

6.3.1 Weighted Neighbourhood Average

A simple way of combining the outputs of the training users' inferrers is by using a *weighted neighbourhood average* (WNA) that weights the inferences according to each training user's similarity to the target user (user similarity measures are introduced in Section 6.4):

$$\hat{r}_{uq} = \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \tilde{r}_{u'q}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \tag{6.1}$$

where $\hat{r}_{uq}$ is the overall inferred polarity for text q by user u, $\mathcal{N}_u$ is the set of user u's neighbours (a subset of the set of training users, as discussed below), $w_{uu'}$ is a non-negative weight assigned to each neighbour u', and $\tilde{r}_{u'q}$ is the polarity inferred by the inferrer of u' for q.[4]

[4]In general, it is possible to extend Equation 6.1 to support negative weights. This requires summing over the absolute values of the weights in the denominator, and ensuring that $\hat{r}_{uq}$ is within the valid polarity range (e.g., in the case of a single neighbour u' with weight $w_{uu'} = -1$, we would get $\hat{r}_{uq} = -\tilde{r}_{u'q}$, which can be resolved by adjusting the $\tilde{r}_{u'q}$ values of negatively-weighted neighbours). We avoided these complications by using only non-negative weights, since high values are more informative than low values in most of the similarity measures we consider (Section 6.5.4). For example, if user u' uses similar language to the target user u, u''s inferrer may be useful for u. However, if u''s language use is completely different from u's, it does not mean that u''s inferrer returns ratings that are *opposite* from those that should be inferred from u's texts, which is the relation that would be encoded by negative weights.

An advantage of MIMU-WNA over SISU is that it does not require any knowledge about the target user. Specifically, when we know nothing about the target user, $\mathcal{N}_u$ is set to include all the training users, and for each neighbour u' we set $w_{uu'} = 1$.

An advantage of MIMU-WNA over SIMU is that if we do have some information about the target user, it can be used to potentially improve performance by calculating similarities to the training users (Section 6.4), and using these similarities to weight the neighbours' inferrers. These assigned weights can also be used to exclude dissimilar users from the neighbourhood $\mathcal{N}_u$, by either setting a static threshold on the number of users or on the magnitude of the similarity values, or by learning such thresholds dynamically for each target user. Learning a threshold dynamically consists of performing cross validation on the target user's labelled texts to find the threshold that minimises the root mean squared error (RMSE) out of a set of candidate thresholds.[5]

6.3.2 Normalised Neighbourhood Inferences

A well-established result from research on neighbourhood-based rating prediction is that normalising the neighbours' predictions can yield substantial improvements over using a simple weighted average of un-normalised predictions (Section 2.5.1). Inspired by this result, we propose two MIMU variants that use the *normalised neighbourhood inferences* (NNI). These variants were outlined in (Herlocker et al., 1999) for rating prediction. We adjusted them for polarity inference by replacing the actual ratings given by the training users (which are available in the case of rating prediction) with inferred polarities (which are available in polarity inference). The variants are defined as follows:

- MIMU-NNI (mean deviation):

$$\hat{r}_{uq} = \mu_u + \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \left(\tilde{r}_{u'q} - \mu_{u'}\right)}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \tag{6.2}$$

- MIMU-NNI (z-score):

$$\hat{r}_{uq} = \mu_u + \sigma_u \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \left(\tilde{r}_{u'q} - \mu_{u'}\right) / \sigma_{u'}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \tag{6.3}$$

where μ_u and σ_u are user u's polarity mean and standard deviation respectively, calculated over u's labelled texts.

[5] In our experiments, we used five-fold cross validation, which makes it possible to learn from relatively small sets of labelled target user texts. We chose to minimise the RMSE because this is the measure that we use for evaluation (Section 3.2), but other measures, such as the mean absolute error, can also be used.

An advantage of MIMU-NNI over MIMU-WNA is that MIMU-NNI accounts for different users having different interpretations of the polarity scale by considering each inferred polarity's deviation from the user's polarity mean (Equation 6.2) or z-score (Equation 6.3), rather than the raw inferred polarity. Both equations use the target user's mean μ_u as the base inference, but Equation 6.3 also accounts for the spread in user polarities. This is done by giving the neighbourhood an overall weight of σ_u (meaning that the neighbourhood would have a small effect on the final inference for target users with a low standard deviation, and vice versa), and weighting each training user u''s inference according to $1/\sigma_{u'}$ (meaning that training users with low standard deviations have a large effect on the final inference, and vice versa).

A disadvantage of MIMU-NNI compared to MIMU-WNA is that MIMU-NNI requires some labelled target user texts to estimate μ_u and σ_u. Hence, MIMU-NNI cannot produce inferences for target users with no labelled texts, and its inferences for target users with only a few texts are likely to be of low quality. In addition, MIMU-NNI's performance for target users with many labelled texts may not improve beyond a certain point because of the dominance of the mean μ_u in the overall inferred polarity $\hat{r}_{uq}$.

6.3.3 Employing the Target User's Inferrer

The inspiration for MIMU-NNI came from neighbourhood-based rating prediction, where the information that is known about the users often includes only their ratings for items. In supervised polarity inference, we do not have to know anything about the items the texts were written about, but by definition, we always have some labelled texts. In case we have enough labelled texts by the target user to build an inferrer, we can augment the inferences made by the *target user's inferrer* (TUI) in a similar manner to MIMU-NNI, replacing μ_u in Equations 6.2 and 6.3 with $\tilde{r}_{uq}$:

- MIMU-TUI (mean deviation):

$$\hat{r}_{uq} = \tilde{r}_{uq} + \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \left(\tilde{r}_{u'q} - \mu_{u'}\right)}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \tag{6.4}$$

- MIMU-TUI (z-score):

$$\hat{r}_{uq} = \tilde{r}_{uq} + \sigma_u \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \left(\tilde{r}_{u'q} - \mu_{u'}\right) / \sigma_{u'}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \tag{6.5}$$

An advantage of this approach over MIMU-NNI is that it has the potential to yield performance that is comparable to SISU's performance even when many labelled texts by the target user are available. This is because the estimator μ_u in MIMU-NNI does not account for the content of the text q, while MIMU-TUI's (and

SISU's) $\tilde{r}_{uq}$ estimator is based on an analysis of the target user's labelled texts – an analysis that is more likely to benefit from having many labelled texts than a simple mean (μ_u).

A possible disadvantage of simply replacing μ_u with $\tilde{r}_{uq}$ is that this may yield poor performance if the TUI was trained only on few labelled texts. Hence, we suggest another TUI variant that is based on MIMU-WNA rather than on MIMU-NNI:

$$\hat{r}_{uq} = \omega_u \tilde{r}_{uq} + (1 - \omega_u) \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \tilde{r}_{u'q}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \tag{6.6}$$

where $\omega_u \in [0, 1]$ controls the relative weight of the target user inferrer compared to the neighbourhood inferrers. When $\omega_u = 0$, the inferred polarity $\hat{r}_{uq}$ is the same as MIMU-WNA's, and when $\omega_u = 1$, the inferred polarity is the same as SISU's.

The value of ω_u is expected to depend on the reliability of the TUI's inferences, which in turn depends on the number of labelled texts used to train the inferrer. Hence, it can either be set statically as a function of the number of labelled texts, or learned dynamically based on cross validation of the target user's labelled texts. The results of our experiments with both variants are reported in Section 6.5.3.

For the static weighting variant – MIMU-TUI (static weight) – we set:

$$\omega_u = \frac{|\mathcal{Q}_u|}{\tilde{\omega} + |\mathcal{Q}_u|} \tag{6.7}$$

where $\mathcal{Q}_u$ is the set of the target user's labelled texts and $\tilde{\omega} > 0$ is a smoothing factor that is set empirically (we used $\tilde{\omega} = 100$ based on preliminary experiments). This allows MIMU-TUI (static weight) to return the same inference as MIMU-WNA for target users with no labelled texts, and gradually increase the weight of the SISU component according to the number of labelled target user texts.

For the dynamic weighting variant – MIMU-TUI (dynamic weight) – we use cross validation over $\mathcal{Q}_u$ to find for each test fold the $\omega_u \in [0, 1]$ that minimises the squared error (or equivalently, the RMSE) over the fold's texts, and set ω_u to its mean over all the folds. Finding the minimising ω_u simply requires solving a quadratic equation since the squared error $\sum_{q \in \mathcal{Q}} (r_{uq} - \hat{r}_{uq})^2$ over the set of the test fold's texts $\mathcal{Q}$ can be rearranged as:

$$\begin{aligned} & \omega_u^2 \sum_{q \in \mathcal{Q}} \left(\tilde{r}_{uq} - \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \tilde{r}_{u'q}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \right)^2 + \\ & \omega_u \sum_{q \in \mathcal{Q}} (-2) \left(\tilde{r}_{uq} - \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \tilde{r}_{u'q}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \right) \left(r_{uq} - \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \tilde{r}_{u'q}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \right) + \\ & \sum_{q \in \mathcal{Q}} \left(r_{uq} - \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \tilde{r}_{u'q}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}} \right)^2 \end{aligned} \tag{6.8}$$

It is worth noting that in principle, it is possible to apply the weighting idea to the MIMU-TUI (mean deviation) and MIMU-TUI (z-score) variants. However, this cannot be done by assigning a weight ω_u to the TUI component $\tilde{r}_{uq}$ and a weight $(1-\omega_u)$ to the neighbourhood component ($\frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \left(\tilde{r}_{u'q} - \mu_{u'}\right)}{\sum_{u' \in \mathcal{N}_u} w_{uu'}}$ or $\sigma_u \frac{\sum_{u' \in \mathcal{N}_u} w_{uu'} \left(\tilde{r}_{u'q} - \mu_{u'}\right)/\sigma_{u'}}{\sum_{u' \in \mathcal{N}_u} w_{uu'}}$ respectively), because the neighbourhood component is not of the same order of magnitude as the TUI component. This can potentially be addressed by assigning two separate weights, one for the TUI component and another for the neighbourhood component – an enhancement that is left for future work.

6.4 User Similarity Measures

In this section, we introduce several user similarity measures, which we use to enhance the inferences made by the MIMU variants introduced in Section 6.3. All the measures output a similarity weight, $w_{uu'} \in [0, 1]$, where u and u' are users (high values indicate a high level of similarity and vice versa). The weights are symmetric, i.e., $w_{uu'} = w_{u'u}$ for all user pairs.

It is worth noting that the similarity measures are not strictly required by our MIMU variants. When similarity cannot be calculated (e.g., for users that we know nothing about), all MIMU variants can revert to employing *equal weights* (EQW) by setting $w_{uu'} = 1$ for all user pairs. However, when similarity can be calculated, it provides MIMU with a way to rank and weight the training users according to their similarity to the target user. The ranking allows MIMU to select the most similar neighbours based on static or dynamic thresholds (Section 6.3.1), and the weighting is used by MIMU to give higher weights to inferrers of users who are similar to the target user than to inferrers of users who are less similar (Equations 6.1, 6.2, 6.3, 6.4, 6.5 and 6.6).

The measures we introduce here explore four different aspects of user similarity:

1. Explicit polarity at the document level, as expressed by ratings assigned by the users to their texts (Sections 6.4.1 and 6.4.2).
2. Implicit polarity at the sentence level, as expressed by the inferred positive-sentence percentage (Section 6.4.3).
3. Interests and style, as indicated by raw token use (Section 6.4.4).
4. Interests and style, as indicated by topic modelling (Section 6.4.5).

Table 6.2 presents a summary of the similarity measures. It specifies each measure's name, the aspect that the measure compares, the equation used to calculate the similarity weight, and the information required for this calculation.

While we cannot cover all the possible ways of measuring user similarity in this study, we hope that the aspects we selected would help shed light on the utility of different similarity measures in our MIMU framework. Moreover, as all the measures

Measure	Compared Aspect	Equation	Required Information
IPV	Item-based polarity vector	6.9	Items with explicit ratings
PRD	Polarity rating distribution	6.10	User texts with explicit ratings
PSPD	Positive-sentence percentage distribution	6.11	User texts
RVP	Raw vocabulary token presence	6.12	User texts
RVF	Raw vocabulary token frequency	6.13	User texts
AT	AT author topic distribution	6.14	User texts
DADT	DADT author topic distribution	6.14	User texts

Table 6.2: Similarity measures

summarise similarity with a single number, we cannot expect them to be exhaustive. However, for the purposes of our MIMU framework, what we need is something that points us in the right direction in terms of ranking and weighting. Most of the performance gains in comparison to the baselines are expected to come from the MIMU framework itself (because it addresses discordances between sentiment-bearing texts by different users), rather than from the similarity measures, which can be seen as the "icing on the MIMU cake".

6.4.1 Baseline: Item-based Polarity Vector

Our baseline measure, *item-based polarity vector* (IPV) similarity, comes from neighbourhood-based rating prediction, where user similarity is commonly based on polarity rating vectors of co-rated items (i.e., items that were rated by both users) (Section 2.5.1). Hence, this measure can be used only when dealing with labelled texts such as movie or product reviews, where we know what items the reviews are about and we are given their explicit polarity ratings, as assigned by the users.

While there are many ways to compare rating vectors, the two measures most commonly used in rating prediction are cosine similarity and the Pearson correlation coefficient (Adomavicius and Tuzhilin, 2005). In preliminary experiments we found that on our datasets, cosine similarity consistently outperformed Pearson correlation (where negative correlation weights were set to zero). Hence, we only used cosine similarity for the experiments presented in Section 6.5.

The cosine similarity between two rating vectors of co-rated items is defined as follows:

$$w_{uu'} = \frac{\boldsymbol{r}_{uu'} \cdot \boldsymbol{r}_{u'u}}{|\boldsymbol{r}_{uu'}||\boldsymbol{r}_{u'u}|} \tag{6.9}$$

where $\boldsymbol{r}_{uu'}$ is user u's vector of ratings for items co-rated with u', sorted in ascending order by item index (note that in general it is likely that $\boldsymbol{r}_{uu'} \neq \boldsymbol{r}_{u'u}$).

IPV accounts for similarities in taste between users, and thus it may be suitable for polarity inference. However, IPV does not measure language similarity, which may be important for polarity inference. Also, it may yield poor performance when there are only a few co-rated items, and is undefined when there are no co-rated items (in which case we set $w_{uu'} = 0$).

6.4.2 Polarity Rating Distribution

The *polarity rating distribution* (PRD) similarity measure defines similarity between users u and u' as one minus the Hellinger distance between their rating distributions:

$$w_{uu'} = 1 - \sqrt{\frac{1}{2} \sum_{r=r_{\min}}^{r_{\max}} \left(\sqrt{\text{prd}(u,r)} - \sqrt{\text{prd}(u',r)} \right)^2} \tag{6.10}$$

where $\text{prd}(u,r)$ denotes the percentage of u's labelled texts with polarity rating r. Ratings are chosen by the author from a discrete scale: $\{r_{\min} = 1, 2, \ldots, r_{\max}\}$.[6]

PRD accounts for the relative positivity or negativity of the users. For instance, if one user mostly gives low ratings and another mostly high ratings, they are considered dissimilar. For PRD to be reliable, we may need a sufficiently large sample of ratings for the two users, which accurately represents their overall rating distributions.

6.4.3 Positive-sentence Percentage Distribution

Positive-sentence percentage (PSP) was defined by Pang and Lee (2005) as the percentage of positive sentences out of the subjective sentences in a document. To detect the subjective sentences, they used the method described in (Pang and Lee, 2004), and to find the positive sentences, they trained a classifier on their dataset of labelled sentences. When used to model *document* similarity in Pang and Lee's (2005) multi-way polarity inference framework, PSP outperformed cosine similarity of token vectors.

Here we introduce a *user* similarity measure based on PSP – *positive-sentence percentage distribution* (PSPD) similarity. PSPD extends the PRD measure by replacing ratings with PSPs. In contrast to Pang and Lee (2005), we define PSP as the percentage of positive sentences among *all* the sentences in a document (rather

[6] Any discrete rating scale can be transformed to a scale that starts from $r_{\min} = 1$.

than just subjective sentences). This generalises the PSP definition to include any type of text, such as message board posts.

PSPD is defined in a similar way to the PRD measure (Section 6.4.2), as one minus the Hellinger distance between the PSP distributions of users u and u':

$$w_{uu'} = 1 - \sqrt{\frac{1}{2}\sum_{k=1}^{K}\left(\sqrt{\text{pspd}(u,k)} - \sqrt{\text{pspd}(u',k)}\right)^2} \tag{6.11}$$

where K is a discretisation factor that is determined experimentally (we set $K = 100$ based on preliminary experiments), and $\text{pspd}(u,k)$ is the percentage of user u's texts with PSPs in the range $\left[\frac{k-1}{K}, \frac{k}{K}\right)$ for $k \neq K$ and $\left[\frac{K-1}{K}, 1\right]$ for $k = K$.

An advantage of PSPD over PRD is that it does not require explicit ratings by the users. However, it is a rather crude measure that may be too noisy to be reliable in some cases. Nonetheless, Pang and Lee's (2005) successful use of PSP for polarity inference leads us to conjecture that PSPD may perform well, at least in some scenarios.

6.4.4 Raw Vocabulary Use

We consider two measures that compare the *raw vocabulary* used by the users:

- RVP, which uses the Jaccard coefficient to compare token *presence*:

$$w_{uu'} = \frac{|\mathcal{V}_u \cap \mathcal{V}_{u'}|}{|\mathcal{V}_u \cup \mathcal{V}_{u'}|} \tag{6.12}$$

  where $\mathcal{V}_u$ is the set of tokens that appear in user u's documents (out of a vocabulary of V tokens).

- RVF, which uses cosine similarity to compare token *frequency*:

$$w_{uu'} = \frac{\boldsymbol{v}_u \cdot \boldsymbol{v}_{u'}}{|\boldsymbol{v}_u||\boldsymbol{v}_{u'}|} \tag{6.13}$$

  where $\boldsymbol{v}_u$ is a length-V vector whose elements are token-occurrence frequencies in user u's documents.

We considered both presence and frequency because, while they are related, presence is expected to be more relevant to polarity inference than frequency (Pang et al., 2002), and frequency is expected to be important in capturing authorship style (since, as discussed in Section 2.3.2, stopwords are likely to be used by all authors, but with varying frequencies according to author style). Both measures are expected to yield a crude representation of user interests. For example, words like "zombie" and "vampire" are more likely to appear in documents written by a horror movie fan than in documents written by a person who likes historical dramas. In

addition, these measures might capture some of the overall positivity or negativity of the users (e.g., the vocabulary of a mostly-positive user probably contains more positive words than the vocabulary of a mostly-negative user) – this can be seen as related to their authorship style, since users are expected to express their positive or negative sentiments in various ways. A potential disadvantage of these measures is that they are calculated over the raw vocabulary and thus may underperform on large sets of noisy documents – we attempt to address this limitation with our topic-based measures (Section 6.4.5).

6.4.5 Topic-based

Our last two similarity measures employ the AT and DADT topic models (Chapter 4) to build compact representations of the users as distributions over author topics,[7] and then compare the users by defining similarity as one minus the Hellinger distance between the distributions:

$$w_{uu'} = 1 - \sqrt{\frac{1}{2}\sum_{t=1}^{T^{(A)}}\left(\sqrt{\theta_{ut}^{(A)}} - \sqrt{\theta_{u't}^{(A)}}\right)^2} \tag{6.14}$$

where $T^{(A)}$ is the number of author topics, and $\boldsymbol{\theta}_u^{(A)}$ is user u's author topic distribution, as defined in Chapter 4. In cases where several estimates of the topic distributions are available due to Gibbs sampling, $w_{uu'}$ is calculated as the mean over all these estimates.

Like the raw vocabulary measures presented in Section 6.4.4, topic-based measures are expected to capture users' style and interests. We hope that the compact representation of users as topic distributions would help handle the inherent noisiness of large datasets of user-generated texts without losing much information, as it did on the authorship attribution task (Chapter 5).

6.5 Evaluation

In this section, we evaluate our MIMU approaches and similarity measures (introduced in Sections 6.3 and 6.4 respectively) by testing their performance on the IMDb62 and IMDb1M datasets (Section 3.3) in comparison to the baselines (Section 6.2). We start by describing our experimental setup (Section 6.5.1), and then proceed to present the results of our experiments on the IMDb62 dataset in Sections 6.5.2, 6.5.3, and 6.5.4. We wrap up the evaluation in Section 6.5.5, where we test the best-performing methods from the IMDb62 experiments on the larger user population of the IMDb1M dataset.

[7] We used only AT and DADT and not the other two models presented in Section 4.2.3 (LDA and AT-FA), because LDA and AT-FA yielded relatively poor performance in authorship attribution scenarios with many authors (Section 5.3).

6.5.1 Experimental Setup

Our experimental setup depends on the dataset. In our experiments on IMDb62, we employed the GivenX protocol, and in our IMDb1M experiments, we employed stratified ten-fold cross validation (Section 3.1). All the experiments were repeated with five different random seeds. In all cases, we report the overall root mean squared error (RMSE) (Section 3.2) of the inferred polarities compared to the actual ratings assigned by the users to their reviews.

Since IMDb62 contains 62 users with 1,000 movie reviews each, employing the GivenX protocol makes it possible to test our methods under relatively controlled conditions by varying the number of labelled target user texts. We focus our attention on the following cases:

- **Given0.** The target user has no labelled texts. In this case, SIMU is used as a baseline (Section 6.2). Since we found that it is too computationally expensive to train SIMU on many labelled texts, in this case we test the effect of setting a cap on the number of labelled *training* user texts (Section 6.5.2).
- **Given5–100.** The number of labelled texts by the target user is small to medium (we experimented with GivenX values of 5, 10, 25, 50 and 100). In this case, SISU is used as a baseline (Section 6.2). We do not cap the number of labelled training user texts in this case because neither SISU nor our MIMU approach exhibits the same runtime issues as SIMU – they remain practical to run even with large amounts of training data.
- **Given200–900.** Many labelled texts by the target user are available (we experimented with GivenX values of 200, 300, 500, 700 and 900). These experiments were run under the same conditions as the Given5–100 experiments. Hence, we plot Given5–100 and Given200–900 results together, but use two separate x-axis scales because the RMSEs of all methods tend to cover a broader spectrum in the Given5–100 range than in the Given200–900 range (e.g., see Figure 6.1).

While our experiments on IMDb62 do not cover all the possible combinations of caps on the labelled training user texts and GivenX values, the presented results still give a good idea about the performance of our MIMU approach in comparison to the SIMU and SISU baselines in various scenarios. To further strengthen the conclusions from the IMDb62 experiments, we ran experiments on IMDb1M, where the number of labelled texts varies from user to user (Section 3.3.2). Hence, the GivenX protocol was not needed in the IMDb1M experiments, and we followed the stratified ten-fold cross validation protocol. This setup represents a challenging scenario that is less controlled than the IMDb62 setups, since about 45% of the texts in each IMDb1M test fold were written by users with less than five labelled texts in the training set.

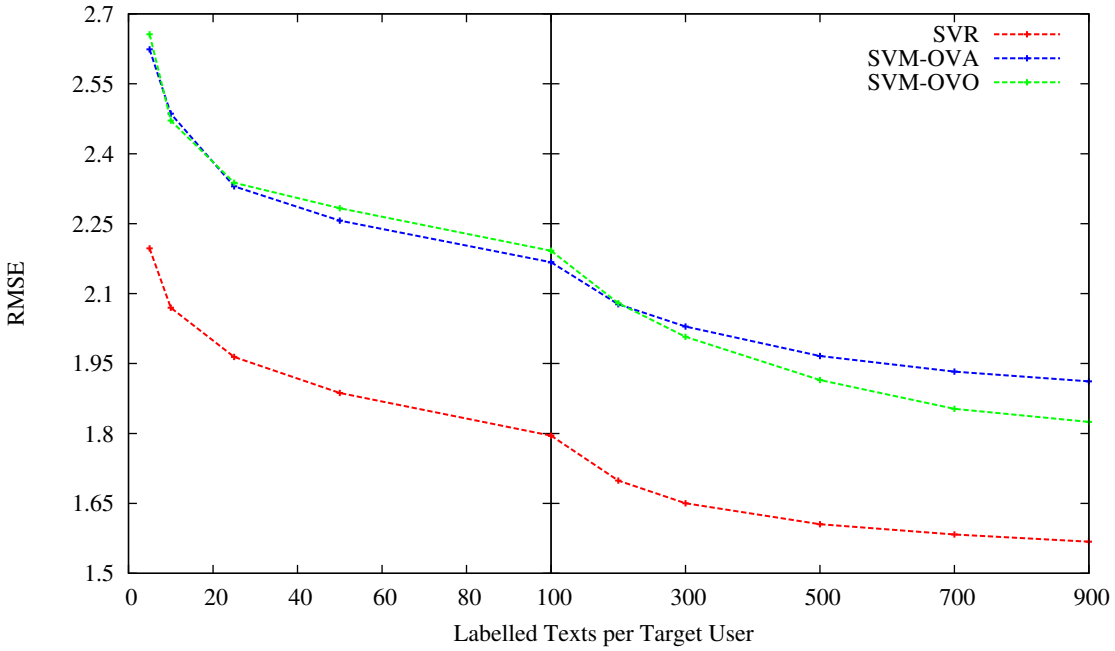


Figure 6.1: SISU experiment results (Given5–900, dataset: IMDb62)

6.5.2 Establishing Baselines

In Section 6.2, we hypothesised that the performance of SISU will be suboptimal when the number of training texts is small, and that SIMU will underperform when trained on labelled texts by many different users. We test these hypotheses in this section by running experiments on the IMDb62 dataset, and thereby also establish baselines for subsequent experiments.

SISU

Since all the polarity inference approaches we consider can employ any type of inference model, we first need to decide on the type of base inferrer to use. Since support vector methods have been shown to yield good polarity inference performance (Section 2.4.1), we tested three variants: (1) binary support vector machines in a one-versus-one setup (SVM-OVO); (2) binary support vector machines in a one-versus-all setup (SVM-OVA); and (3) support vector regression (SVR). Based on previous results by Pang et al. (2002) that we reproduced in preliminary experiments, we chose to use unigram presence as features for all inferrers (rather than unigram frequencies). We employed these three inferrers in a SISU setup with the GivenX protocol, with X values that range from 5 to 900.

Figure 6.1 shows the results of this experiment. Unsurprisingly, SVR yielded better results than both SVM-OVO and SVM-OVA, since SVR takes the ordering of the 10 possible polarity labels into account, while SVMs see the polarity labels as classes with no ordinal relations. While both SVM setups performed comparably when the number of labelled texts was small, SVM-OVO tended to perform better

than SVM-OVA when many labelled texts were available.[8] This is probably because some of SVM-OVO's 45 sub-classifiers were trained on very few texts when the number of labelled texts was small, and thus benefitted more from the addition of labelled texts than SVM-OVA's 10 sub-classifiers. Since SVR clearly outperformed the other inferrers, we used it in all subsequent experiments.

The results of this experiment support our hypothesis regarding SISU's performance. As expected, the overall RMSE decreased as the number of texts used for training increased, independently of the base inferrer.[9] However, the rate of improvement in performance decreased to the point where adding hundreds of texts had only a minor effect on the RMSE.

In Section 6.2, we also hypothesised that using a random inferrer of a different user from the target user would yield poor performance. We verified this hypothesis by running a Given0 experiment, where for each test text a different inferrer is selected randomly from the training users' inferrers (each of these inferrers was trained on 1,000 labelled texts). As expected, the RMSE in this case was quite high (2.336) – even higher than simply using the global polarity mean over all the labelled texts (which yielded an RMSE of 2.241). The RMSE in the SISU Given5 case was lower (2.197), which indicates that even if there are only five labelled texts by the target user, it may be better to use SISU than to pick a random inferrer of a different user (or use the global mean), even if that other inferrer was trained on many labelled texts.

SIMU

Unlike SISU, SIMU can employ labelled texts by users other than the target user. SIMU achieves this by simply ignoring any information about the users who wrote the texts, and building a single inferrer based on all the labelled texts (Section 6.2.1). We tested SIMU's performance by running a Given0 experiment where training texts were sub-sampled in a stratified manner according to their authors, varying the sampled percentage from 0.5% to 30%. We stopped at 30% because we found that it is too computationally expensive to run SIMU on the full training set (it would have taken *weeks* of CPU time per fold), and the best performance was achieved with small sample sizes.[10]

[8] All the differences between SVR and the two SVM setups are statistically significant. The differences between the two SVM setups are statistically significant only for GivenX values of 50, 100 and 300–900.

[9] These decreases in RMSE are unlikely to be due to the GivenX protocol, as the RMSE of using the global polarity mean over all the labelled texts was about 2.24 for all GivenX values.

[10] Although the runtime issue might be due to the implementation of SVR that we used, it still demonstrates that SIMU could be problematic to use in practice. In addition, we did not encounter this problem in experiments with our MIMU approach, even when the full training dataset was used (Section 6.5.3).

Percentage of Training Texts	RMSE
0.5%	**2.101**
1%	**2.088**
2.5%	**2.091**
5%	2.124
10%	2.174
20%	2.261
30%	2.317

Table 6.3: SIMU experiment results (Given0, dataset: IMDb62)

Table 6.3 presents the results of this experiment (the best statistically significant results are highlighted in boldface). As expected, adding training texts yielded worse performance after a certain point, probably because SIMU finds it hard to generalise due to discordances between labelled texts by different users. This limitation of SIMU is especially apparent in comparison to SISU, which achieved similar performance to the best SIMU approach with only 10 training texts by the target user (Figure 6.1).

6.5.3 Comparison of MIMU Variants

In this section, we test the MIMU variants introduced in Section 6.3. We compare the MIMU variants to each other and to the baselines by employing the same experimental setups as in Section 6.5.2. At this stage we are interested in the standalone performance of the MIMU approaches, and thus we do not consider user similarity in this section. Hence, we set the training user weights $w_{uu'}$ to 1 for all users u and u' (denoted EQW – equal weights).

MIMU-WNA versus the Baselines

In our first set of experiments we compared MIMU-WNA to the baselines. Since MIMU-WNA with EQW effectively returns an unweighted average of the neighbours' inferences (Section 6.3.1), it does not require any information about the target user. Hence, we could compare it to both SISU, which is trained only on target user texts, and to SIMU, which does not distinguish between texts by different users. As discussed in Section 6.5.1, we perform the comparison to SISU under the Given5–900 setups, and the comparison to SIMU under the Given0 setup.

Figure 6.2 presents the results of an experiment that compares MIMU-WNA to SISU. As the figure shows, MIMU-WNA yielded improvements over SISU only in cases where the number of labelled texts was small.[11] This is not surprising, since

[11] All the differences between MIMU-WNA and SISU are statistically significant, except for the Given25 and Given50 cases.

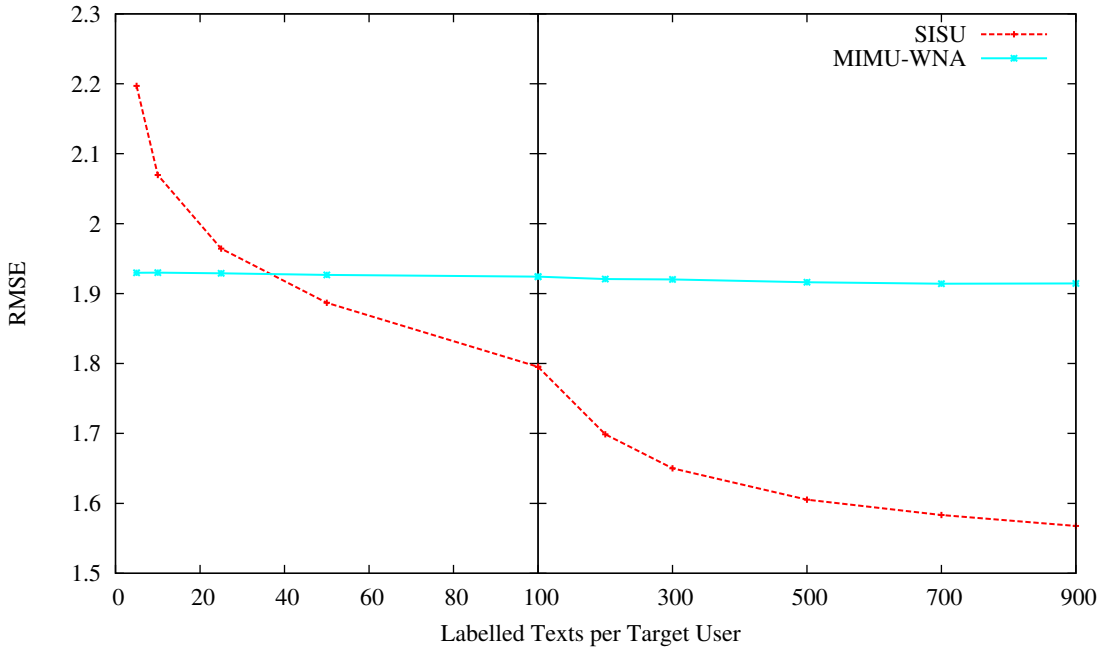


Figure 6.2: MIMU-WNA versus SISU (Given5–900, dataset: IMDb62)

MIMU-WNA with EQW does not take any information about the target user into account. We show improvements over this result with the other MIMU variants in subsequent experiments.

It is worth noting that even though the number of labelled target user texts was not expected to affect MIMU-WNA with EQW, MIMU-WNA's performance got slightly better as labelled target texts were added (as a close look at Figure 6.2 reveals). We believe that this occurred because of the way the GivenX protocol is implemented – the number of labelled texts is changed for *all* the target users in a given fold, and each target user also serves as a training user for the other target users (Section 3.1). For example, in the Given5 case there were six or seven users per fold with 5 labelled texts in the training set, while in the Given900 case these users had 900 labelled texts. Hence, it is unsurprising that MIMU-WNA's inferences were somewhat better in the latter case, where about 10% of the base inferrers are trained on more texts (and thus should be of higher quality).[12] Nonetheless, this minor issue does not affect our conclusions since we compared all the methods under the same conditions.

Figure 6.3 presents the results of an experiment that compares MIMU-WNA to SIMU.[13] As the figure shows, MIMU continuously improved as *training* user texts were added, compared to SIMU that reached its best performance with about 10

[12] A possible way of addressing this issue is by performing leave-one-out cross validation on the users rather than ten-fold cross validation, but this would cause a considerable increase in the number of experiments that need to be run.

[13] All the differences between the two methods for each training percentage are statistically significant.

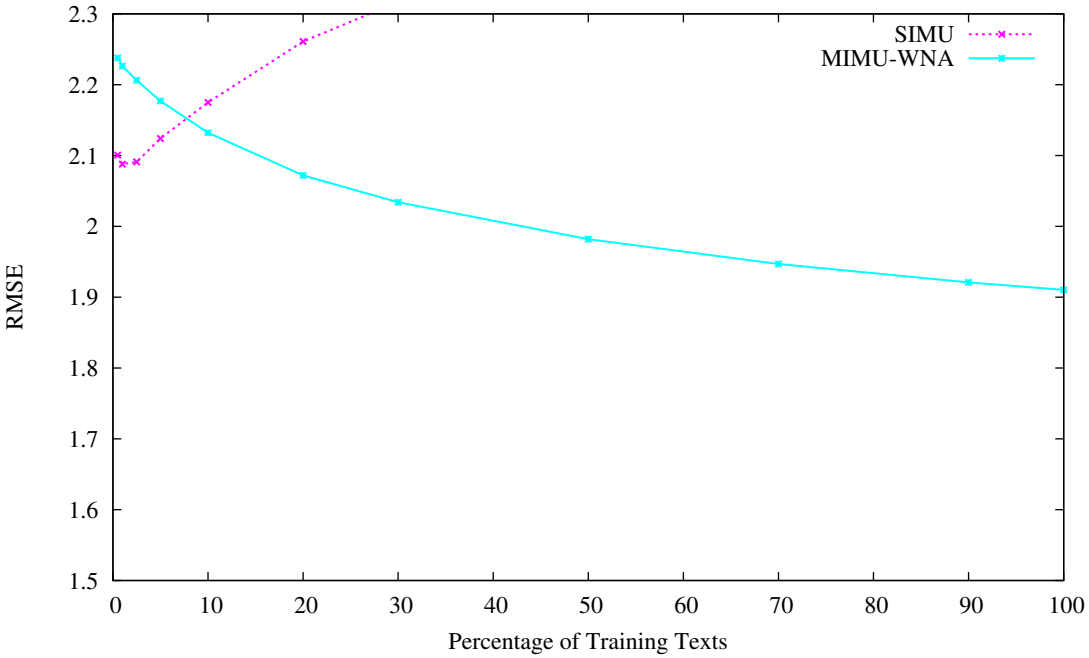


Figure 6.3: MIMU-WNA versus SIMU (Given0, dataset: IMDb62)

texts per training user. In addition, running MIMU took well under an hour of CPU time even on the full training dataset – much less than SIMU, which we did not run because it would have taken weeks.

MIMU-NNI Variants

Our two MIMU-NNI variants require some labelled texts by the target user to calculate the user's mean polarity (for both variants) and standard deviation (for the z-score variant) (Section 6.3.2). Hence, we only experimented with setups where some labelled target user texts are available (Given 5–900), and compared the performance of MIMU-NNI to MIMU-WNA and SISU. Figure 6.4 presents the results of this experiment.

The results obtained with both MIMU-NNI variants were virtually identical, with slightly better results for the mean deviation variant when few labelled target user texts were available, and lower RMSEs with the z-score variant when more labelled target user texts were added.[14] This is probably because z-score normalisation uses the standard deviation of the given polarities, which requires more labelled texts to be reliably estimated than the polarity mean.

Another notable result is that MIMU-NNI yielded improvements over MIMU-WNA in the Given10–900 range, and also outperformed SISU when up to about 100 labelled target user texts were available.[15] This is in line with what we expected,

[14] All the differences between the MIMU-NNI variants are statistically significant, except for the Given25 case.

[15] All the differences between the MIMU-NNI variants and MIMU-WNA or SISU are statistically significant, except for the Given5 case for MIMU-WNA versus either of the MIMU-NNI variants.

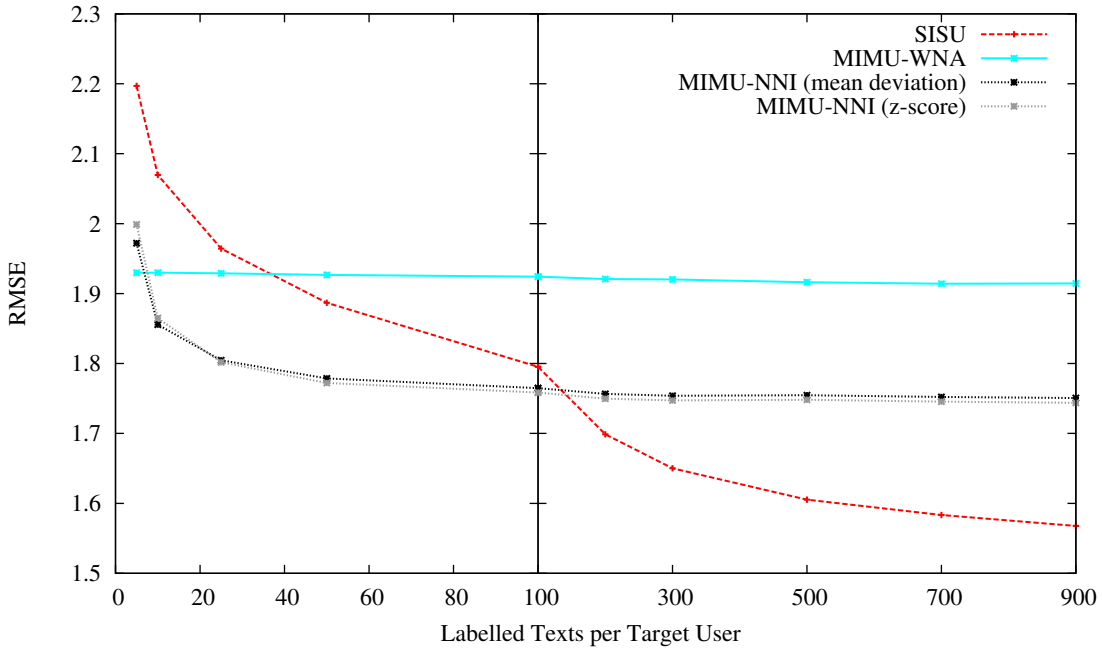


Figure 6.4: MIMU-NNI versus SISU and MIMU-WNA (Given5–900, dataset: IMDb-62)

since MIMU-NNI integrates some knowledge about the target user with the inferences obtained from the neighbourhood, while MIMU-WNA with EQW takes no advantage of information about the target user.

Finally, the results of this experiment highlight two potential weaknesses of the MIMU-NNI approach. First, both MIMU-NNI variants yielded higher RMSEs than MIMU-WNA in the Given5 scenario, probably because the estimates for the mean and standard deviation of the target user polarities were unreliable in this case (but the differences between the MIMU-NNI variants and MIMU-WNA were not statistically significant in this case). Second, MIMU-NNI's performance virtually plateaued in the Given100–900 range, while SISU's kept improving. This is probably because the estimates of the mean and standard deviation became stable in this range. As discussed in Section 6.3.3, these weaknesses of MIMU-NNI can be addressed by taking the target user inferrer (TUI) into account with our MIMU-TUI variants, which are considered in the rest of this section.

MIMU-TUI Variants

In Section 6.3.3 we proposed four different ways of integrating the target user u's inferrer (TUI) into the MIMU framework. The first two variants replace MIMU-NNI's target user mean with TUI's inferred polarity. The other two variants combine TUI's inference with the neighbourhood inferences, according to a weight ω_u (Equation 6.6), which is either set statically according to user u's number of labelled texts, or learned dynamically based on cross validation. The Given5–900 results obtained with these four variants are presented in Figure 6.5.

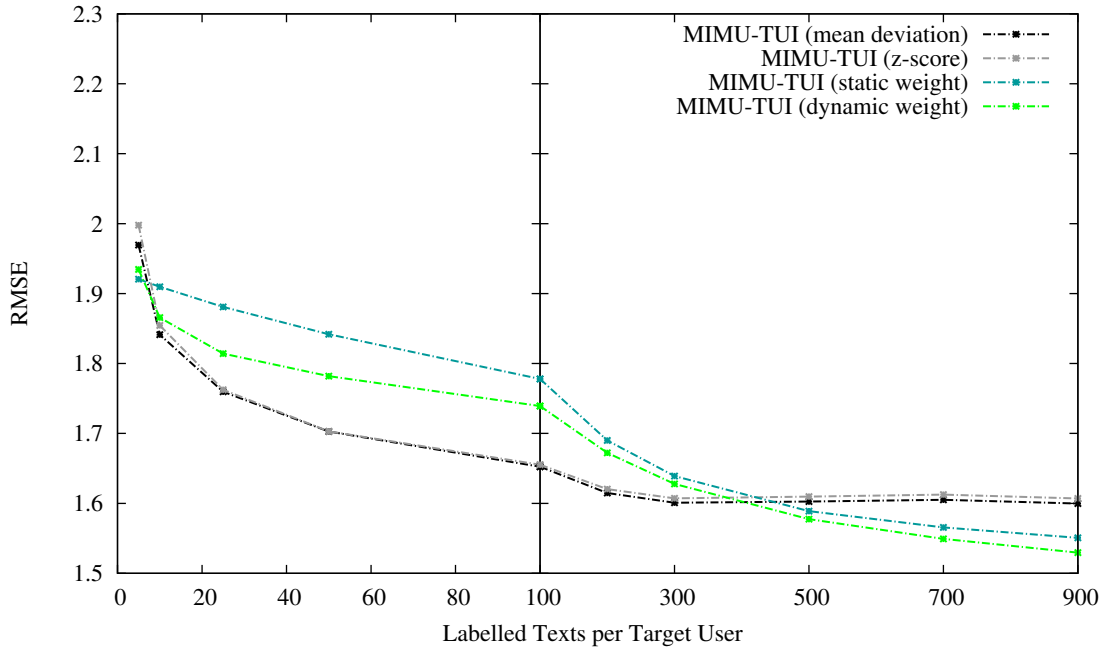


Figure 6.5: Comparison of MIMU-TUI variants (Given5–900, dataset: IMDb62)

As Figure 6.5 shows, there were only small differences in performance between MIMU-TUI (mean deviation) and MIMU-TUI (z-score), though there was a small advantage to MIMU-TUI (mean deviation) for most GivenX values.[16] This is similar to the small difference between the MIMU-NNI (mean deviation) and MIMU-NNI (z-score) results. By contrast, the differences between MIMU-TUI (static weight) and MIMU-TUI (dynamic weight) were larger, with dynamic weighting consistently outperforming static weighting in cases where at least 10 labelled texts were available.[17] This is unsurprising, because the dynamic weighting approach actively utilises the labelled texts, as the dynamic weight is automatically learned for each target user using cross validation, rather than set statically based only on the number of labelled target user texts.

When comparing MIMU-TUI (dynamic weight) to MIMU-TUI (mean deviation), there is no clear advantage to any variant. MIMU-TUI (dynamic weight) performed better on very low or very high GivenX values, while MIMU-TUI (mean deviation) yielded the best performance for mid-range GivenX values.[18] For low GivenX values, this may be because MIMU-TUI (mean deviation) always uses the TUI component, while the dynamic weight variant may give a low weight to the TUI component in these cases and rely mostly on the neighbourhood component. For mid-range

[16]The differences between MIMU-TUI (mean deviation) and MIMU-TUI (z-score) are statistically significant in all cases except for Given25, Given50 and Given100.

[17]The differences between MIMU-TUI (static weight) and MIMU-TUI (dynamic weight) are statistically significant in all cases except for Given5.

[18]The differences between MIMU-TUI (dynamic weight) and MIMU-TUI (mean deviation) are statistically significant in all cases.

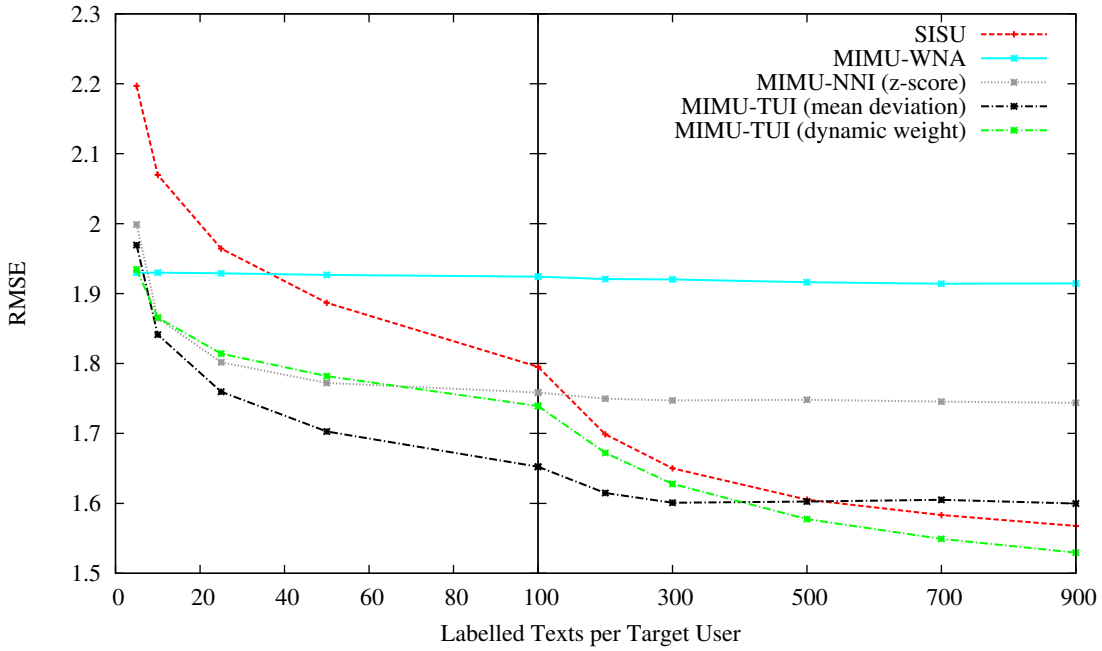


Figure 6.6: MIMU-TUI versus SISU, MIMU-WNA and MIMU-NNI (Given5–900, dataset: IMDb62)

GivenX values, the fact that MIMU-TUI (dynamic weight) does not normalise the neighbours' inferences may make the contribution of the neighbourhood component more noisy than in the MIMU-TUI (mean deviation) case. Finally, for high GivenX values, MIMU-TUI (dynamic weight) may decrease the neighbourhood component's weight due to the higher reliability of the TUI component, and thus the effect of the noisiness of the neighbourhood component is smaller. This suggests that a MIMU-TUI variant that combines the strengths of both MIMU-TUI (dynamic weight) and MIMU-TUI (mean deviation) may potentially obtain better results than either method. Experiments with such a variant are left for future work.

Figure 6.6 compares the two best MIMU-TUI variants to SISU, MIMU-WNA and MIMU-NNI (z-score).[19] These results demonstrate the advantage of our MIMU approach over the SISU baseline, as MIMU-TUI (dynamic weight) outperformed SISU for all GivenX values. In addition, since the best-performing MIMU-TUI variants either outperformed both MIMU-WNA and MIMU-NNI or performed comparably to them, we can conclude that either of the MIMU-TUI variants should be used in practice when at least five labelled texts by the target user are available.

[19] All the differences between the MIMU-TUI variants and the other methods are statistically significant, except for MIMU-TUI (mean deviation) versus MIMU-WNA in the Given5 case and versus SISU in the Given500 case; and MIMU-TUI (dynamic weight) versus MIMU-WNA in the Given5 case and versus MIMU-NNI (z-score) in the Given10, Given25 and Given50 cases.

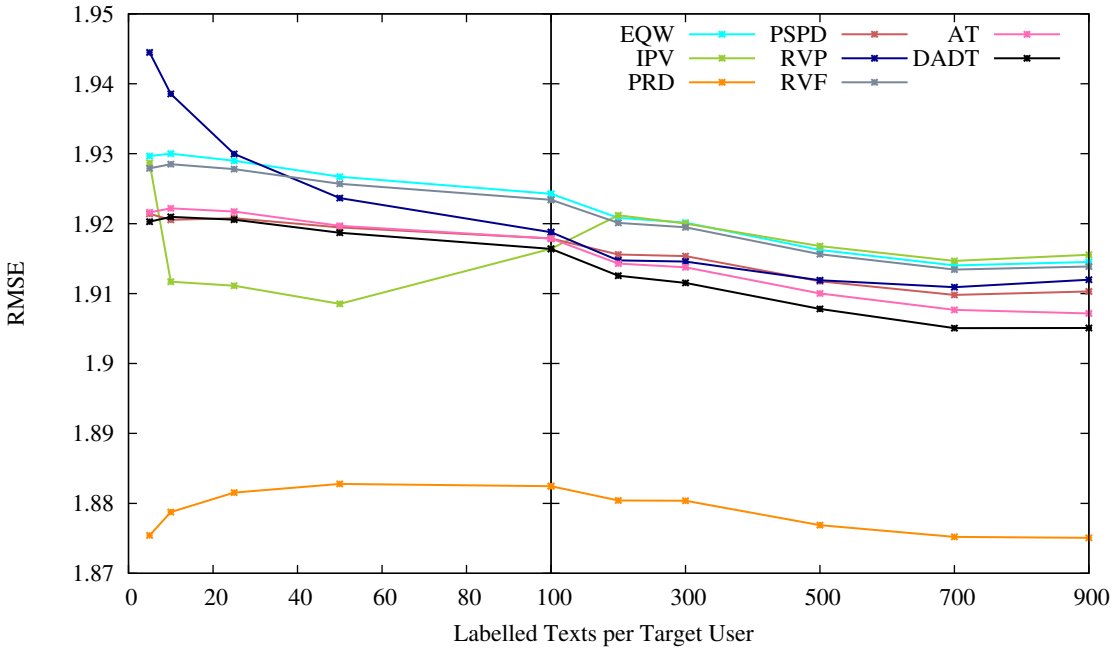


Figure 6.7: MIMU-WNA with similarities and without thresholds (Given5–900, dataset: IMDb62)

6.5.4 Comparison of Similarity Measures

In this section, we report the results of IMDb62 experiments that test the similarity measures introduced in Section 6.4. We first present the results of experiments with MIMU-WNA normalisation (Section 6.3.1), which returns only the weighted average of the neighbourhood inferences, thereby allowing us to test the net effect of employing the similarity measures separately from estimates based on the target user (which are used by MIMU-NNI via the target user's mean and by MIMU-TUI via the target user's inferrer). Then, we show the results obtained with the two best normalisation approaches from Section 6.5.3: MIMU-TUI (mean deviation) and MIMU-TUI (dynamic weight), when used together with the similarity measures.

MIMU-WNA experiments

Given5–900. Figures 6.7 and 6.8 present the results of MIMU-WNA experiments that compare the effect of using no similarity measure (i.e., *equal weights* – EQW) and using the baseline *item-based polarity vector* (IPV) similarity measure to the measures that we defined in Section 6.4, when at least some labelled texts by the target user are available. We ran these experiments under two different setups: (1) without thresholds on the number of nearest neighbours (Figure 6.7); and (2) with dynamic thresholds that were learned separately for each target user, as explained in Section 6.3.1 (Figure 6.8). Note that the figures use different y-axis ranges from those used in previous sections because the RMSEs obtained here span smaller ranges than in previous experiments.

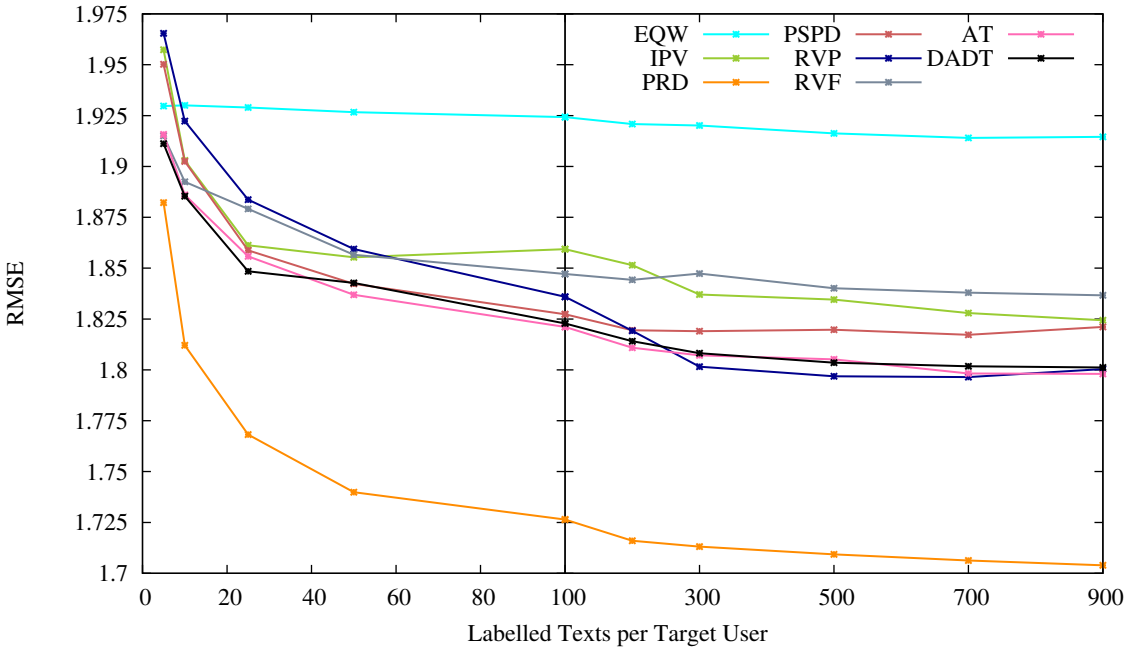


Figure 6.8: MIMU-WNA with similarities and dynamic thresholds (Given5–900, dataset: IMDb62)

As Figure 6.7 shows, the effect of the *weighting* aspect of the similarity measures was rather small, and only the *polarity rating distribution* (PRD) measure stands out with the best performance. By contrast, allowing the *ranking* implied by the measures to influence the inferences via dynamic thresholding (Figure 6.8) increased the margins between the measures, with PRD still yielding the best results.[20] This leads us to conclude that choosing the "right" neighbours is of greater importance than the exact weights given to the chosen neighbours (Equation 6.1). We can also conclude that at least under MIMU-WNA, overall rating behaviour is the most important aspect for polarity inference among the aspects tested by our similarity measures. This may be because MIMU-WNA does not normalise the inferences of the neighbours, and therefore does not directly account for the different interpretations of the polarity scale and the varying levels of positivity of the users. Employing the PRD measure allows MIMU-WNA to account for these differences.

Turning our attention to the similarity measures other than PRD, we see that except for the RVP measure in the Given5 case (with or without thresholds) and the Given10 case (without thresholds), the results obtained with the similarity measures were better than or comparable to the results obtained with EQW.[21] This may be because all the similarity measures have some relation to the polarity inference

[20]Without thresholds, all the differences between PRD and the other measures are statistically significant. With thresholds, all the differences between PRD and the other measures are statistically significant, except for RVF, AT and DADT in the Given5 case.

[21]Without thresholds, all the differences between EQW and the similarity measures are statistically significant except for IPV in the Given5, Given10 and Give200–900 cases, and RVP in the Given25 case. With thresholds, all the differences between EQW and the similarity measures are

task. Moreover, when dynamic thresholds are applied, each target user's neighbourhood can be optimised when enough labelled texts are available. Hence, even if the ranking imposed by the similarity measure is "wrong" (i.e., dissimilar users are ranked highly), the dynamic threshold can reduce the effect of the ranking by choosing a large neighbourhood size. Therefore, when dynamic thresholds are used, performance that is at least as good as EQW's performance is virtually guaranteed for high GivenX values, regardless of the similarity measure used. Nonetheless, the observable differences between the similarity measures are still interesting. Specifically:

- The *item-based polarity vector* (IPV) baseline measure yielded the lowest RMSE of the non-PRD measures only in the Given10–50 cases without thresholds, but did not significantly outperform the other non-PRD measures in any case.[22] This may be because in general, similar interests – as expressed by explicit ratings – are not necessarily indicative of language similarity, which is more related to the polarity inference task (e.g., two users may like the same movies but use completely different language to express their opinions). In addition, we suspect that IPV's performance in the Given10–Given50 cases without thresholds is due to it being the only measure that is likely to yield some similarity values of zero – when the sets of co-rated items are empty (on a side note, PRD's performance in the Given5 case without thresholds may also be due to a similar reason). Hence, unlike the other measures, IPV caused users to be filtered out from the neighbourhood even when no thresholds were used. As demonstrated by the results obtained with thresholds, such filtering is likely to improve performance. This effect was probably most pronounced in the Given10–50 cases because for higher GivenX values most training users had at least some items in common with the target user (so not many users were filtered out), while for lower GivenX values too few training users had items in common with the target user (so too few users were included in the neighbourhood).

- The differences between the *positive-sentence percentage distribution* (PSPD) measure and the other non-PRD measures are most apparent in Figure 6.8, which shows that PSPD's best results were obtained in the Given10–300 range with thresholds.[23] PSPD's poorer performance in the Given5 case may be explained by lack of information from which to calculate the PSPs, while in

statistically significant except for IPV, PSPD, RVF, AT and DADT in the Given5 case, and IPV, PSPD and RVP in the Given10 case.

[22] All the differences between IPV and the top non-PRD, non-IPV measure are statistically significant in all cases except for Given5–100 without thresholds, and Given10–50 with thresholds.

[23] With thresholds, the differences between PSPD and the best-performing non-PRD measure are statistically significant in all cases except for in the Given10–300 range.

the Given500–900 cases it may be because the PSP distribution estimates stabilise beyond a certain number of labelled texts.

- The *raw vocabulary presence* (RVP) and *raw vocabulary frequency* (RVF) measures complemented each other in terms of performance, with RVF performing better than RVP for low GivenX values, and RVP performing better than RVF for high GivenX values.[24] For low GivenX values this may be because RVP, which considers token presence, may discard too much information when a small number of labelled texts is available, while RVF utilises this small amount of data better because it relies on continuous token frequency, which is more informative than binary presence. On the other hand, RVP outperformed RVF for high GivenX values probably because of the de-noising effect of considering presence rather than frequency.
- The AT model and DADT model similarity measures performed comparably to each other when thresholds were used, though DADT outperformed AT by a small margin in the Given25 and Given100–900 cases when no thresholds were used. In addition, AT and DADT performed better than or comparably to both RVP and RVF.[25] It is worth noting that in order to keep the number of experiments at a feasible level, we did not tune AT's and DADT's parameters. Instead, we used the settings that yielded the best authorship attribution performance on the IMDb62 dataset (Section 5.3.4). It appears that in this case DADT's approach of de-noising the author representations by modelling authors and documents over two disjoint sets of topics is of little benefit in comparison to AT's approach of using only author topics. This may appear to stand in contrast to the results of our authorship attribution experiments (Chapter 5), but it could be because the similarity measures use the soft clustering aspect of the topical user models, and thus do not require the models' full discriminatory power, which is where DADT's strengths lie (Section 4.3.3). Nonetheless, we are encouraged by the fact that employing topical user models seems to combine the strengths of both the RVP and RVF measures, as AT and DADT obtained the best results of the token-based measures that aim to capture user interests and style, despite the fact that these models operate in a space of lower dimensionality than the raw vocabulary measures.

[24]The differences between RVP and RVF are statistically significant in all cases except for Given25–100 with thresholds.

[25]The differences between DADT and AT are *not* statistically significant in all cases except for Given25 and Given100–900 without thresholds. The differences between AT and RVP are statistically significant in all cases except for Given100–300 without thresholds and Given100–900 with thresholds. The differences between DADT and RVP are statistically significant in all cases except for Given100–200 without thresholds and Given50 and Given200–900 with thresholds. The differences between either AT or DADT and RVF are statistically significant in all cases except for Given5 and Given10 with thresholds.

Similarity	Threshold	RMSE
EQW	—	1.920
PSPD	38	1.900
RVP	23	**1.878**
RVF	25	1.906
AT	19	1.891
DADT	20	1.889

Table 6.4: MIMU-WNA with similarities (Given0, dataset: IMDb62)

Given0. Table 6.4 presents the results of this experiment in the special case where the target users have no available texts with polarity labels (Given0), but have authored texts of a different type (message board posts, which are available in IMDb62). In this case, we could not apply a dynamic threshold for each user because the application of such a threshold requires labelled texts by the target user. Therefore, we set a global static threshold on the number of neighbours, presenting the results obtained with the best-performing threshold. Only similarity measures that can employ unlabelled texts could be used in this case (i.e., all the measures except for PRD and IPV).

As Table 6.4 shows, using similarity to filter users yielded better results than not using any similarity measure (EQW), with the best results obtained by RVP, closely followed by DADT and AT. While the fact that RVP outperformed RVF may seem surprising, since the opposite was true for positive low-range GivenX values (Figure 6.8), this may be attributed to the fact that in the Given0 setup, most target users have many texts, because we excluded users with no posts at all from the test set. These post-less users were included in the test set for Given5 and above. Hence, our observation that RVP tends to perform poorly on users with only a few texts still holds. The other results are also consistent with those obtained with non-zero GivenX values. Specifically, PSPD's performance in comparison to the other measures is in line with the results obtained with PSPD for high GivenX values. In addition, the differences between DADT and AT are not statistically significant, which strengthens the observation we made above regarding the effect of soft clustering under MIMU. Moreover, the fact that in general, the topical user modelling approach yielded results that are close to using RVP is a further indication that the models we consider effectively capture user interests and style. We will put this result to use in our rating prediction model (Chapter 7), which requires low-dimensional representations of the users and thus cannot employ raw vocabulary representations.

Method	**Similarity**	**Labelled Texts per Target User**									
		5	10	25	50	100	200	300	500	700	900
SISU	—	2.197	2.070	1.964	1.887	1.795	1.699	1.650	1.605	1.583	1.568
MIMU-WNA	EQW	1.930	1.930	1.929	1.927	1.924	1.921	1.920	1.916	1.914	1.915
	PRD	**1.882**	**1.812**	1.768	1.740	1.726	1.716	1.713	1.709	1.706	1.704
MIMU-TUI (mean deviation)	EQW	1.970	1.841	1.760	1.703	1.652	1.615	1.601	1.603	1.605	1.600
	IPV	2.030	1.857	1.759	1.701	1.646	1.607	1.588	1.577	1.570	1.557
	PRD	1.972	**1.831**	**1.730**	**1.672**	**1.624**	1.591	1.580	1.577	1.570	1.559
	PSPD	1.996	1.842	1.742	**1.675**	**1.621**	**1.586**	**1.574**	1.571	1.566	1.556
	RVP	2.002	1.852	1.746	1.681	1.625	**1.585**	**1.572**	1.570	1.567	1.556
	RVF	1.997	1.843	1.746	1.683	1.632	1.596	1.579	1.576	1.570	1.558
	AT	1.984	**1.826**	**1.735**	**1.672**	**1.621**	**1.588**	1.577	1.573	1.567	1.558
	DADT	1.975	**1.827**	**1.731**	**1.674**	**1.620**	1.589	1.577	1.574	1.567	1.558
MIMU-TUI (dynamic weight)	EQW	1.934	1.866	1.814	1.782	1.739	1.672	1.628	1.577	1.549	1.530
	IPV	1.958	1.854	1.789	1.769	1.733	1.671	1.627	1.577	1.548	1.527
	PRD	1.935	1.841	1.771	1.733	1.698	1.647	1.610	**1.563**	**1.534**	**1.513**
	PSPD	1.952	1.860	1.791	1.754	1.713	1.657	1.617	1.568	1.539	1.520
	RVP	1.964	1.879	1.808	1.765	1.713	1.650	1.610	**1.563**	1.537	1.517
	RVF	1.925	1.841	1.793	1.756	1.718	1.660	1.619	1.570	1.542	1.522
	AT	1.911	1.839	1.781	1.746	1.703	1.647	1.609	**1.562**	**1.535**	**1.515**
	DADT	**1.901**	1.837	1.780	1.749	1.705	1.649	1.611	**1.563**	**1.535**	**1.515**

Table 6.5: MIMU-TUI with similarities (Given5–900, dataset: IMDb62)

MIMU-TUI experiments

Although considering user similarity in isolation from other factors by employing MIMU-WNA had a positive effect on performance, we saw in Section 6.5.3 that the best MIMU-TUI variants outperformed MIMU-WNA in most cases. Hence, to wrap up the similarity-based experiments, we ran the two best-performing MIMU variants – MIMU-TUI (mean deviation) and MIMU-TUI (dynamic weight) – together with the similarity measures and dynamic thresholds on the number of neighbours (Section 6.3.1). The results of this experiment are summarised in Table 6.5, with the best statistically significant results for each GivenX value highlighted in boldface.

In terms of differences between the similarity measures, the results of this set of experiments are in line with the observations we made based on the results of the MIMU-WNA experiments in the Given5–900 range. One notable exception is that the PRD measure did not outperform the other measures when used jointly with MIMU-TUI. This may be because PRD fills the role of finding training users that are similar to the target user in terms of levels of positivity, which may be of lesser importance under MIMU-TUI than under MIMU-WNA, since the target user's inferrer inherently handles individual positivity levels.

As the results show, the performance gains obtained by considering similarity with MIMU-TUI were not as large as in the MIMU-WNA experiments. This is not surprising, as MIMU-TUI already handles the most relevant information to polarity inference by taking the target user inferrer into account when generating inferences. Nonetheless, we are encouraged by the fact that considering user similarity had a positive effect on performance, as we expected it to be the "icing on the MIMU cake" (Section 6.4), and it further supports our hypothesis that users should be taken into account when performing polarity inference.

6.5.5 Experiments with a Large User Population

In this section, we report the results of our experiments on the IMDb1M dataset (Section 3.3.2), in which we tested the best-performing methods from the IMDb62 experiments. The IMDb1M dataset represents a more varied user population than the IMDb62 dataset, since the number of users in IMDb1M is much larger than in IMDb62, and the number of available labelled texts varies from user to user in IMDb1M. Hence, as discussed in Section 6.5.1, we employed ten-fold cross validation rather than the GivenX protocol.

We had to make a few small adjustments to the methods to enable running them on the IMDb1M dataset:

- As in the IMDb62 experiments (Section 6.5.2), we could not train SIMU on the full training set. Hence, the presented RMSE for SIMU was obtained by training on only 2.5% of the available texts (this percentage yielded the best results).
- About 45% of the texts in each test fold were written by users with less than five labelled texts in the training set (Section 6.5.1). Hence, methods that depend on labelled target user texts (SISU, MIMU-WNA with PRD or DADT, and the two MIMU-TUI variants) were assigned a *fallback method* to use when the number of labelled target user texts is under a given threshold. This threshold was set to five labelled target user texts, because our IMDb62 experiments showed that at around Given5 considering the labelled target user texts starts yielding reasonable results. We employed either SIMU or MIMU-WNA with EQW as fallback methods.
- Since most training users have few labelled texts, we also set a threshold on the number of labelled texts for a training user's inferrer to be included in the neighbourhood for all MIMU variants. We set this threshold to 200 labelled texts, based on the results of the IMDb62 MIMU-WNA versus SIMU experiment (Figure 6.3), where MIMU-WNA and SIMU yielded comparable performance when at least 200 labelled texts were used to train each of MIMU's base inferrers.

Method	Fallback Method	Similarity	RMSE
SIMU	—	—	2.518
SISU	SIMU	—	2.500
MIMU-WNA	MIMU-WNA (EQW)	EQW	2.721
		PRD	2.644
		DADT	2.678
MIMU-TUI (mean deviation)	MIMU-WNA (EQW)	EQW	2.562
		PRD	2.559
		DADT	2.556
	SIMU	EQW	2.447
		PRD	2.444
		DADT	**2.441**
MIMU-TUI (dynamic weight)	MIMU-WNA (EQW)	EQW	2.583
		PRD	2.574
		DADT	2.581
	SIMU	EQW	2.469
		PRD	2.460
		DADT	2.466

Table 6.6: IMDb1M experiment results

The results of this experiment are presented in Table 6.6, which shows the RMSE for each combination of method, fallback method and similarity measure (where applicable). As the table shows, the best result was obtained by the MIMU-TUI (mean deviation) method, with the SIMU fallback method and the DADT similarity measure.[26] It is unsurprising that MIMU-TUI (mean deviation) outperformed MIMU-TUI (dynamic weight), since the IMDb62 experiments showed that MIMU-TUI (mean deviation) yielded better performance in the Given10–400 range, which is where most of the IMDb1M users with five or more labelled texts lie. In addition, a possible explanation for the differences in performance between PRD and DADT is that PRD was of more benefit when used with MIMU-WNA and MIMU-TUI (dynamic weight) because these variants do not take user interpretations of the polarity scale into account, while PRD's benefit diminished in the MIMU-TUI (mean deviation) case because the neighbours' polarity scales are accounted for by the mean deviation component (Equation 6.4).

As the results show, choosing the right fallback method is important because of the large number of users with few labelled texts. SIMU proved to be a better fallback method than MIMU-WNA with EQW, probably because the number

[26]DADT was run with the same parameter settings used for the IMDb1M authorship attribution experiments (Section 5.3.4).

of training users with enough labelled texts to be used by MIMU-WNA was too small (21 or 22, depending on the training fold). This may also explain why using similarity did not yield large improvements, except for in the MIMU-WNA case, which is also in line with the IMDb62 results (Section 6.5.4). Nonetheless, we are encouraged by the fact that our MIMU-TUI approach yielded the best results when combined with SIMU as the fallback method.

While the RMSE improvements obtained with our MIMU approach over the baselines are smaller for IMDb1M than for IMDb62, they still support our main hypothesis – that users should be taken into account when analysing sentiment in their texts. The IMDb1M dataset with its large number of users with few labelled texts is a challenging testbed for user-aware polarity inference methods. However, it cannot be seen as representing the full spectrum of data encountered in "real life". For example, there are situations where many texts by each user exist, but they are not tagged for polarity (e.g., emails and social media messages, which can be analysed if user consent is given). A possible avenue for future research is to harness such texts in a semi-supervised setting, e.g., by labelling them for polarity using a simple method such as SIMU, and then using the automatically-labelled texts to train one of our MIMU-TUI variants, which are expected to yield good performance when many labelled texts are available.

6.6 Summary and Conclusions

In this chapter, we introduced an approach to polarity inference that takes users into account when inferring sentiments from their texts. In our experiments, our approach outperformed two baselines – one that does not take users into account and another that does – in a variety of scenarios based on our two IMDb datasets. Our experimental results provide empirical evidence for the connection between users and sentiments expressed in their texts, and show that our approach successfully harnesses this connection to improve polarity inference performance.

More specifically, our *Multiple Users, Multiple Inferrers* (MIMU) approach comprises three aspects that enable integration of different types of information about the users (Section 6.3). First, information about who wrote the labelled training texts is used to build user-specific inferrers (MIMU-WNA) (Section 6.3.1). Second, information about who wrote the text whose polarity we want to infer enables us to tailor MIMU's inferences to a specific target user (MIMU-NNI and MIMU-TUI) (Sections 6.3.2 and 6.3.3). Third, information from similarity measures enables us to weigh the inferences made by MIMU's base inferrers (Section 6.4). As our evaluation showed, employing each of these aspects makes MIMU's inferences increasingly accurate.

Our main goal in employing our similarity measures was to explore aspects of user similarity that pertain to polarity inference (Section 6.4). As such, we did not attempt to devise the ultimate similarity measure, but instead compared representative measures for each of the aspects we considered. However, one encouraging result was that of the text-based measures, those based on topical user models (Chapter 4) yielded the best performance in most cases. This serves as further affirmation that topic models yield useful representations of users based on their texts.

The work presented in this chapter, which is one of the first studies on harnessing user information in polarity inference, may be extended in several ways (Section 8.2). For example, as noted by Li et al. (2011) in a study that followed the initial publication of our MIMU-based results in (Seroussi et al., 2010), MIMU may underperform when only few labelled texts by the users are available. Li et al. addressed this limitation by introducing a method based on tensor factorisation that takes into account users, products and review texts (in a manner inspired by matrix factorisation approaches to rating prediction, which we discuss in Chapter 7). However, Li et al.'s method is specifically tailored to polarity inference of product reviews, unlike our MIMU approach, which can handle any type of text. Nonetheless, it is encouraging to see other researchers who recognise the need to consider users when performing polarity inference, as establishing this need was our main goal in this chapter.

In the next chapter, we move on from user identity and its influence on polarity inference to the rating prediction task, where the goal is to *predict* users' future sentiments, rather than *infer* the sentiments they have already expressed. As rating prediction has a long history of taking users into account (this is required by the definition of the task), our contribution in Chapter 7 is in the direction of bringing awareness of user-generated texts into rating prediction techniques. This can be seen as complementary to the current chapter: here, we addressed polarity inference in a user-aware manner by introducing an approach inspired by rating prediction methods, while the next chapter addresses rating prediction in a text-aware manner by drawing inspiration from polarity inference and authorship attribution.

Chapter 7

Text-aware Rating Prediction

Chapter 5 investigated topical user models in the context of authorship attribution, while Chapter 6 explored the connection between users and sentiments expressed in their texts. In this chapter, we address the problem of predicting user sentiment in the form of ratings, focusing on *new users* who submitted only a few or no ratings. Our main contribution is in showing that considering texts by such users by employing topical user models yields personalised rating predictions that are more accurate than predictions yielded by baselines that rely only on ratings.

Recent years have seen a growing interest in the collaborative rating prediction task (Section 2.5). In its most basic form, the task is to predict the rating a target user would give to a target item given past ratings by the target user and by other users. Rating prediction is often an important part of recommendation generation, where a recommender system has to choose items that will be of interest to users. It has been shown that even small improvements in the accuracy of rating predictions may lead to better recommendations (Koren, 2008).

Despite the interest in the rating prediction task, the issue of generating accurate predictions for new users is often overlooked (Section 2.5). This is due, in part, to the experimental setup commonly used to test rating prediction methods – predictive accuracy is often evaluated using cross validation over ratings, which gives equal weights to, for example, errors on a user with 100 training ratings and 100 test ratings and errors on 100 users with one training rating and one test rating each. This setup is used despite the fact that users with few ratings often form the majority of the population (Section 7.4.1). The prevalence of this setup means that many of the techniques developed in recent years work well for users with many ratings, but may perform poorly for users with few ratings.

Some of the most accurate techniques for rating prediction are based on *matrix factorisation* (MF) (Koren et al., 2009). MF techniques for rating prediction reduce the dimensionality of the user-item rating matrix by building a lower-rank representation of the matrix. Each user and item are represented by a small number of

latent factors, and predictions are generated based on the interaction between the user factors and the item factors. While MF techniques produce accurate predictions in many cases, they tend to perform poorly for users with few ratings (Section 7.4.2), and they cannot produce *personalised* predictions for users with no ratings at all.

This chapter addresses the problem of producing personalised rating predictions for new users. We generate personalised rating predictions by extending MF to consider user attributes, and show that our extended model yields improved predictive accuracy compared to traditional MF and to a non-personalised baseline. We study two types of user attributes. First, we consider attributes derived from demographic information that was explicitly supplied by the users. Second, we consider implicit attributes that are inferred from user-generated texts via topical user models (Chapter 4). We find that in both cases, our model obtains an improvement in predictive accuracy over two baselines that consider only ratings. This is an encouraging result, especially in the latter case, as it shows that we can generate personalised rating predictions that are relatively accurate without requiring users to provide explicit ratings or information about themselves.

This chapter is structured as follows. Matrix factorisation and its extended version that considers user attributes are described in Sections 7.1 and 7.2 respectively. Our approach to deriving user attributes from demographic information and from texts are discussed in Section 7.3. Section 7.4 presents and discusses the results of our evaluation, and Section 7.5 concludes the chapter.

7.1 Matrix Factorisation for Rating Prediction

Matrix factorisation (MF) techniques for collaborative rating prediction build a lower-rank representation of the user-item rating matrix, and then use this representation to generate rating predictions (Koren et al., 2009). This section presents the basic MF framework, which serves as a baseline to the work presented in this chapter.

Given a (usually sparse) rating matrix $\boldsymbol{R}_{N\times M}$ by N users for M items, the most basic form of MF builds a rank-F representation of $\boldsymbol{R}$, decomposing it into a user-factor matrix $\boldsymbol{X}_{F\times N}$ and an item-factor matrix $\boldsymbol{Y}_{F\times M}$, such that $\boldsymbol{R} \approx \boldsymbol{X}^\top \boldsymbol{Y}$.[1] The predicted rating for a user u and an item i is calculated as follows:

$$\hat{r}_{ui} = \boldsymbol{x}_{\cdot u}^\top \boldsymbol{y}_{\cdot i} \tag{7.1}$$

where $\boldsymbol{x}_{\cdot u}$ denotes the u-th column of $\boldsymbol{X}$, and $\boldsymbol{y}_{\cdot i}$ denotes the i-th column of $\boldsymbol{Y}$.

[1]Note that while we denote latent factors with the same letters we used to represent latent assignments in the topic model descriptions in Chapter 4, the factors discussed here and the topics discussed in Chapter 4 are unrelated.

This dot product can be seen as modelling the relationship between user preferences ($\boldsymbol{x}_{\cdot u}$) and corresponding item characteristics ($\boldsymbol{y}_{\cdot i}$). For example, if factor f corresponds to historical themes in movies, then x_{fu} is expected to be high for a user u that is interested in history, and y_{fi} is expected to be high for a historical drama i. Hence, the product $x_{fu}y_{fi}$ is expected to be high, meaning that – all other things being equal – a recommender system that is based on such predictions is likely to recommend movie i to user u.

The basic MF model can be enhanced to include user and item biases (Koren et al., 2009), i.e., the tendency of users and items to deviate from the global rating mean. When biases are included, Equation 7.1 becomes:

$$\hat{r}_{ui} = \mu + b_u^{(U)} + b_i^{(I)} + \boldsymbol{x}_{\cdot u}^\top \boldsymbol{y}_{\cdot i} \tag{7.2}$$

where μ is the global rating mean, $\boldsymbol{b}^{(U)}$ (with elements $b_u^{(U)}$) is the vector of user biases, and $\boldsymbol{b}^{(I)}$ (with elements $b_i^{(I)}$) is the vector of item biases.

Typically, many ratings are missing from $\boldsymbol{R}$. This poses a challenge for training the model, which is addressed by learning $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{b}^{(U)}$ and $\boldsymbol{b}^{(I)}$ from the set of known ratings $\mathcal{R}$ by minimising the following objective function (Koren et al., 2009):

$$\begin{aligned} &\sum_{r_{ui} \in \mathcal{R}} \left(r_{ui} - \underbrace{\left(\mu + b_u^{(U)} + b_i^{(I)} + \boldsymbol{x}_{\cdot u}^\top \boldsymbol{y}_{\cdot i} \right)}_{= \hat{r}_{ui}} \right)^2 \\ &+ \lambda_1 \sum_{u=1}^{N} \|\boldsymbol{x}_{\cdot u}\|^2 + \lambda_2 \sum_{i=1}^{M} \|\boldsymbol{y}_{\cdot i}\|^2 + \lambda_3 \|\boldsymbol{b}^{(U)}\|^2 + \lambda_4 \|\boldsymbol{b}^{(I)}\|^2 \end{aligned} \tag{7.3}$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are parameters to the regularisation part of the objective function, which avoids overfitting.

Following Koren et al. (2009), we minimise Equation 7.3 using stochastic gradient descent (Algorithm 7.1).[2] In addition to the set of known ratings $\mathcal{R}$, the number of latent factors F, and the regularisation parameters $\lambda_1, \ldots, \lambda_4$, the input to Algorithm 7.1 includes the initial learning rate γ_0, the learning rate decrease factor λ_γ, and the number of steps to run the algorithm *numSteps*. The algorithm returns the factor matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ and the bias vectors $\boldsymbol{b}^{(U)}$ and $\boldsymbol{b}^{(I)}$, which are used in Equation 7.2 to generate rating predictions.

[2]We chose stochastic gradient descent due to its speed and ease of implementation. *Alternating least squares* (ALS) is another popular technique that can be used to minimise the objective function (Koren et al., 2009). Although ALS implementations can be parallelised and handle non-sparse datasets faster than stochastic gradient descent, these advantages are irrelevant in our case.

Algorithm 7.1 Minimising Equation 7.3 by stochastic gradient descent

Input: $\mathcal{R}, F, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \gamma_0, \lambda_\gamma$, *numSteps*
Output: $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{b}^{(U)}, \boldsymbol{b}^{(I)}$

```
Initialise X, Y, b^(U), b^(I) with random values in [0, 1]
γ ← γ_0
μ ← Σ_{r_ui ∈ R} r_ui / |R|
for step = 1 to numSteps do
    for all r_ui ∈ R do
        e_ui ← r_ui − (μ + b_u^(U) + b_i^(I) + x_{·u}^⊤ y_{·i})
        b_u^(U) ← b_u^(U) + γ (e_ui − λ_3 b_u^(U))
        b_i^(I) ← b_i^(I) + γ (e_ui − λ_4 b_i^(I))
        for f = 1 to F do
            x_fu ← x_fu + γ (e_ui y_fi − λ_1 x_fu)
            y_fi ← y_fi + γ (e_ui x_fu − λ_2 y_fi)
    γ ← λ_γ γ
```

7.2 Matrix Factorisation with User Attributes

In this section, we present our extension to the basic MF model: *Matrix Factorisation with User Attributes* (MFUA). Our main aim in designing the MFUA model was to address the new user problem, since we found that the basic MF model performs poorly when generating predictions for new users, even though such users often form a large part of the population (Section 7.4.2).

Our MFUA model is based on a suggestion made by Koren et al. (2009), who proposed adding an element to Equation 7.2 that takes into account user attributes, such as gender and age. In their model, it is assumed that every user is described by a set of binary attributes that is given to the model as input. An attribute-factor matrix is then learned from the available ratings to model the interactions between user attributes and item factors, and used when generating predictions. This model was proposed as a possible extension of the basic MF model, but it was not evaluated in (Koren et al., 2009). It also was not specifically aimed at alleviating the new user problem.

The focus of this chapter is on improving the accuracy of rating predictions for new users. Hence, we introduce three extensions to Koren et al.'s suggested model: (1) reformulation of the model as a switching model (Burke, 2002); (2) addition of attribute biases; and (3) allowing probabilistic assignment of attributes.

In our model, rating predictions for a target user depend on the number of known ratings by the user. In contrast to Koren et al. (2009), we employ our *attribute-based* model only if the target user submitted less than n ratings (n is set empirically). Otherwise, predictions are generated by the *user-based* model, i.e., using Equation 7.2. Formally, we describe users by a T-dimensional vector of

attribute probabilities, where each element $p(t|u)$ of this vector is the probability that user u has attribute t. The predicted rating of an item i for a user u is:

$$\hat{r}_{ui} = \begin{cases} \mu + b_i^{(I)} + \sum_{t=1}^{T} p(t|u) \left(b_t^{(A)} + \boldsymbol{z}_{\cdot t}^{\top} \boldsymbol{y}_{\cdot i} \right) & |\mathcal{R}_u| < n \\ \mu + b_u^{(U)} + b_i^{(I)} + \boldsymbol{x}_{\cdot u}^{\top} \boldsymbol{y}_{\cdot i} & \text{otherwise} \end{cases} \tag{7.4}$$

where $\boldsymbol{b}^{(A)}$ (with elements $b_t^{(A)}$) is the vector of attribute biases, $\boldsymbol{z}_{\cdot t}$ is the t-th column vector of the attribute-factor matrix $\boldsymbol{Z}_{F \times T}$, and $\mathcal{R}_u$ is user u's known rating set.

We employ a *switching approach* because we observed that employing Equation 7.2 to generate predictions based on the user-based model can lead to poor performance for new users. This is because the rating predictions are partly based on the components $b_u^{(U)}$ and $\boldsymbol{x}_{\cdot u}$, which are learned from very few ratings in the new user case (Algorithm 7.1). In this case, ignoring the user bias and factors, i.e., generating a non-personalised prediction by using $\mu + b_i^{(I)}$, actually improves predictive accuracy (Section 7.4.2). However, generating such non-personalised predictions is roughly equivalent to predicting the item mean. Recommender systems that use such predictions may only recommend the most popular items, which users may already know about, and thus the recommendations will not be very useful. By contrast, switching to the attribute-based model results in the generation of *personalised* predictions even for users with no ratings at all, as long as some information about the users is known. As our experiments show, predictions produced by our attribute-based model are more accurate than the non-personalised predictions produced by using $\mu + b_i^{(I)}$ (Section 7.4).

The *addition of attribute biases* models the item-independent effect of the user attributes, in the same way the user and item biases model the item- and user-independent effects of the user and item ratings, respectively. For example, if users in a certain segment of the population tend to give lower ratings than the rest of the population, this will be captured by the bias elements that describe this population segment.

We allow for *probabilistic assignment of attributes to users* to handle cases where attributes cannot be assigned with absolute certainty. For example, when user-generated texts are available, we employ topical user models (Chapter 4) to generate topic distributions that represent the users, in which case we define an attribute for each topic and $p(t|u)$ represents the probability of user u using topic t according to the underlying model (Section 7.4.4). Another example pertains to cases where user names are known and their gender and age can be inferred based on census data.

An interesting point to note is that the user-based model is likely to have many more parameters to infer than the attribute-based model. Specifically, the user-based model has $N + M + F \times N + F \times M = (F + 1)(N + M)$ parameters, while the

Algorithm 7.2 Minimising Equation 7.5 by stochastic gradient descent

Input: $\mathcal{R}, F, \lambda_5, \lambda_6, \gamma_0, \lambda_\gamma$, *numSteps*, $\boldsymbol{Y}, \boldsymbol{b}^{(I)}$, $p(t|u)$ for all $t \in \{1, \ldots, T\}$ and $u \in \{1, \ldots, N\}$
Output: $\boldsymbol{Z}, \boldsymbol{b}^{(A)}$

```
Initialise Z and b^(A) with random values in [0, 1]
γ ← γ_0
μ ← Σ_{r_ui ∈ R} r_ui / |R|
for step = 1 to numSteps do
    for all r_ui ∈ R do
        e_ui ← r_ui − (μ + b_i^(I) + Σ_{t=1}^{T} p(t|u) (b_t^(A) + z_.t^⊤ y_.i))
        for t = 1 to T such that p(t|u) > 0 do
            b_t^(A) ← b_t^(A) + γ (p(t|u) e_ui − λ_6 b_t^(A))
            for f = 1 to F do
                z_ft ← z_ft + γ (p(t|u) e_ui y_fi − λ_5 z_ft)
    γ ← λ_γ γ
```

attribute-based model has $(F + 1)(T + M)$ parameters. Since typically $T \ll N$, the attribute-based model has considerably fewer parameters. Thus, a switching approach is justified when dealing with users with few ratings, because the user-based model is more prone to overfitting since it has more parameters to infer than the attribute-based model.

We train our model in two stages. First, we minimise Equation 7.3 using Algorithm 7.1 to learn the user-based model. Then, we use Algorithm 7.2 to learn the attribute-based model by minimising Equation 7.5:

$$\sum_{r_{ui} \in \mathcal{R}} \left(r_{ui} - \left(\mu + b_i^{(I)} + \sum_{t=1}^{T} p(t|u) \left(b_t^{(A)} + \boldsymbol{z}_{\cdot t}^\top \boldsymbol{y}_{\cdot i} \right) \right) \right)^2 + \lambda_5 \sum_{t=1}^{T} \|\boldsymbol{z}_{\cdot t}\|^2 + \lambda_6 \|\boldsymbol{b}^{(A)}\|^2 \tag{7.5}$$

where λ_5 and λ_6 are parameters to the regularisation part of the objective function, which avoids overfitting.

Learning the model in two stages ensures that the attribute-based model does not affect the predictions made by the user-based model, because the item-factor matrix and the item biases are considered constant in the second stage. Therefore, they are not modified by Algorithm 7.2. Note that we train both the attribute-based model and the user-based model on all the available ratings to maximise the amount of information available to both models.

7.3 Derivation of User Attributes

Our MFUA model takes as input the vector of attribute probabilities for each user. A question that arises given any type of information about the users is how to encode this information as attribute probability vectors, while taking into account the constraints imposed by both the data and the model. Ideally, we would like the

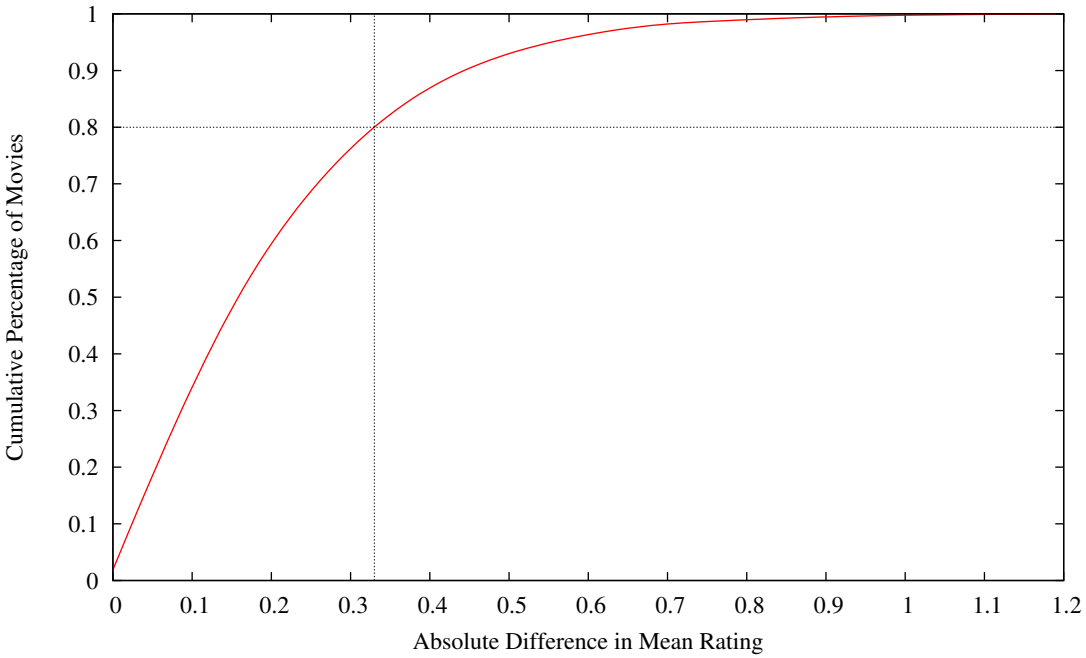


Figure 7.1: Gender differences in rating patterns (dataset: MovieLens)

attributes to capture commonalities between users, while still maintaining a degree of difference between the predictions produced for different users. In addition, we do not want to have too many user attributes, as it may result in overfitting of the attribute-based model (Section 7.2).

We consider two types of user attributes: demographic attributes that are based on information explicitly supplied by the users, and implicit attributes that are inferred from texts written by the users. Section 7.3.1 motivates the use of such attributes in rating prediction, while Sections 7.3.2 and 7.3.3 detail the derivation procedures for demographic and text-based attributes, respectively.

7.3.1 Motivation: Can Attributes Work?

Before moving on to discuss attribute derivation in detail, we pause to present some evidence that indicates that considering attributes makes sense. Specifically, we looked at the differences in per-gender rating means for movies from the MovieLens dataset (Section 3.3.6). To obtain reliable estimates, we only considered movies with at least ten ratings by males and ten ratings by females, which resulted in a subset of 2,634 movies. The cumulative percentage of movies as a function of the absolute difference between the genders' mean ratings is presented in Figure 7.1.

As Figure 7.1 shows, for some items user gender is correlated with user preferences as expressed by ratings. For example, the absolute difference in mean rating between male and female users is at least 0.33 for about 20% of the movies. Anecdotally, we found that among the most popular movies (with at least 100 ratings by

users of each gender), the top movies in terms of rating difference are *Dirty Dancing* (2.96 mean rating by men and 3.79 by women), *Jumpin' Jack Flash* (2.58/3.25), *Dumb & Dumber* (3.34/2.70), and *Grease* (3.37/3.98). The bottom popular movies in terms of absolute difference include *Jerry Maguire* (3.76 by either gender) and *Trainspotting* (3.96). This suggests that demographic attributes such as gender can potentially be harnessed to improve rating prediction accuracy.

Keeping in line with the prominent theme of this thesis (using texts to model users), we remind readers that user-generated texts contain indicators of demographic characteristics, such as gender (Section 2.3.1). Further, indicators that are useful in identifying authors of texts can also be used to discover demographic information about users. This leads us to posit that topical user models, whose applicability to authorship attribution was demonstrated in Chapter 5, could also prove useful as sources of user attributes that capture user demographics (along with other user characteristics and interests).

7.3.2 Demographic Attributes

Even when high-quality demographic information about the users is available, consideration needs to be given to how this information is converted to probabilistic attributes. We explain this conversion process through examples taken from the MovieLens dataset (Section 3.3.6):

- The *gender* characteristic is straightforward to represent, as the vast majority of people define themselves as being either male or female. If we assume that this is the only demographic information we have about the users, we can define two gender attributes: male (attribute 1) and female (attribute 2). If we have complete knowledge about all the users, then for each user u, either $p(t = 1|u) = 1$ and $p(t = 2|u) = 0$ or $p(t = 1|u) = 0$ and $p(t = 2|u) = 1$. We can deal with missing information about gender either by defining an additional attribute to represent users with unknown gender, or by using extraneous information (e.g., user name) to assign probabilities to these attributes, while maintaining the equality $\sum_{t=1}^{2} p\,(t|u) = 1$.
- The *age* characteristic can be represented as a set of attributes, one for each value (i.e., attribute 1 for one-year-olds, attribute 2 for two-year-olds, etc.). However, some ages may be under-represented in the datasets (e.g., training data for 100-year-olds would probably be harder to obtain than data for 30-year-olds), and differences in taste are unlikely to vary much between users whose ages are close (e.g., it is probably safe to assume that the differences between 25-year-olds and 26-year-olds are negligible as a whole). Hence, it is advisable to define only a few attributes that represent discrete age bands.

Such bands may be defined based on observed differences in rating patterns between age groups in the training data, or obtained from an external source. For example, when dealing with movie ratings, one could use the Internet Movie Database's age bands (under 18, 18–29, 30–44, and 45+),[3] those defined by the Motion Picture Association of America (under 12, 12–17, 18–24, 25–39, 40–49, 50–59, and 60+),[4] or the age bands provided in the MovieLens dataset (under 18, 18–24, 25–34, 35–44, 45–49, 50–55, 56+), which we used in our experiments (Section 7.4.3). In all cases, one attribute is defined for each age band. If complete age information is available, then for each user the probability of only one of the age band attributes is set to 1, while the probabilities of the remaining age band attributes are set to 0. If the age information is missing, the age band attributes can be assigned probabilities based on population statistics.

- The *place of residence* characteristic can take many forms. Examples include GPS coordinates, IP address, postcode, or a physical address. Like the age characteristic, the place of residence characteristic requires grouping to obtain useful attributes. For example, the MovieLens dataset contains the United States postcodes for most users, and we converted them to US states to obtain attributes that are shared by at least a few users. We handled unknown or missing postcodes by defining an additional attribute rather than assigning probabilities to the existing state attributes, because users with unknown postcodes may not reside in the US.
- The *occupation* characteristic can be given in a fine-grained or coarse-grained form, like the place of residence. In the MovieLens dataset, 20 different occupations are given, including an unknown/other category. Hence, we simply defined an attribute for each of the occupation categories. However, in cases where hundreds of occupations are given, these can be grouped to form attributes in a similar manner to the groupings used for the age and place of residence characteristics.

While other demographic user information may exist in some scenarios (e.g., income and education levels), the treatment of the characteristics given above demonstrates what should be considered when converting this information into attributes. The two main considerations should be: (1) creation of meaningful user groups that contain enough users to support generalisations by the MFUA model; and (2) handling missing information (which is naturally accounted for by our probabilistic

[3]As presented in the Internet Movie Database's movie-specific rating breakdowns (retrieved from *The Godfather*'s breakdown at `www.imdb.com/title/tt0068646/ratings` on 3 April, 2012).

[4]As presented in the Motion Picture Association of America's *2011 Theatrical Market Statistics* (retrieved from `www.mpaa.org/policy/industry` on 3 April, 2012).

attributes for characteristics like age, where the attributes can be assigned probabilities according to population statistics).

Another important point that arises when many attributes exist is that compact user representations are required to avoid overfitting of the attribute-based model (Section 7.2). In such cases, *Principal Component Analysis* (PCA) (Jolliffe, 2002) may be used to reduce the dimensionality of the attribute space. PCA is a well-established technique for dimensionality reduction. The main idea behind PCA is that variables are often not independent (e.g., the binary variables isPregnant and gender are correlated). Therefore, PCA transforms the original variables into new, uncorrelated, variables (the principal components) that are ordered so that the first principal components account for most of the variance contained in the original data. In our case, we transform the original user attributes into the principal component space and keep only the first few components to obtain a compact representation of the users. The transformed attribute values are discretised by dividing the values into quartiles, and defining one attribute for the values that fall within each quartile (i.e., four attributes for each principal component).[5] Our results indicate that retaining a fairly small number of principal components yields comparable performance to the performance obtained when using all the attributes (Section 7.4.3).

7.3.3 Text-based Attributes

While it is fairly straightforward to integrate explicit demographic attributes into MFUA, it is more challenging to infer attributes from user-generated texts. As discussed in Section 2.3, authorship profiling research shows that simple features such as the occurrence frequencies of the most frequent tokens are indicators of user demographics and personality (Argamon et al., 2009). However, accurate inference of demographics and personality traits typically requires training a model on a domain-specific corpus of labelled texts. Therefore, we chose to infer user attributes directly from the texts, rather than try to infer demographics or personality traits. We considered using token frequencies (by defining an attribute for each token in the vocabulary, i.e., users would be represented as their overall word distribution), but at least a few hundred tokens are often required to effectively profile the users. Thus, using token frequencies directly would result in a user representation that is too sparse for our needs. In addition, using only the most frequent tokens may cause loss of information, as item preferences are likely to be related to users' interests and sentiments, which are partly reflected by their overall vocabulary use.

To overcome the sparsity problem that arises from using token frequencies, we employ the AT and DADT models, defining an attribute for each author topic, i.e., the attribute probabilities $p(t|u)$ from Equation 7.4 are defined as the probability

[5] We experimented with several types of quantiles, but found quartiles to yield the best performance.

of author topic t being allocated to user u. As we showed in Chapters 5 and 6, these two models yield good performance on the authorship attribution task and constitute good sources of user similarity in our polarity inference framework, with DADT usually yielding better authorship attribution performance than AT, and performing similarly to AT in the polarity inference scenarios we considered. Hence, we expect AT and DADT to yield user representations that are helpful for our rating prediction goal.

A point to note is that here we use the topic distributions directly rather than calculating values based on the topics. This is unlike the most successful authorship attribution methods, which calculated the author probabilities based on the inferred models, and unlike our polarity inference approach, which utilised the similarity values based on the topics. In addition, as our Gibbs sampling approach to inferring the topics does not allow us to average different samples due to topic exchangeability (Chapter 4), we use only one sample per model. A possible way of integrating multiple samples is by learning multiple attribute-based models and averaging their predictions, but this may be too computationally expensive in realistic settings. Moreover, since we found that the attribute-based model's performance is stable across different random seeds, we decided to stick with using only one sample.[6]

7.4 Evaluation

In this section, we evaluate the performance of our MFUA approach with demographic and text-based attributes by running experiments on the MovieLens and IMDb1M datasets (Section 3.3). We first describe our experimental setup (Section 7.4.1), and show that MF suffers from the new user problem (Section 7.4.2). Then, we evaluate the performance of MFUA with demographic attributes (Section 7.4.3) and with our text-based attributes (Section 7.4.4).

7.4.1 Experimental Setup

Our focus in this chapter is on rating prediction for new users. Hence, we employ the GivenX protocol in all the experiments (repeated with five different random seeds), focusing on users with very few ratings (the Given0 and Given1 cases). We decided to focus on users with no given ratings or with only one given rating because such users often form the majority of the user population. For example, about 71% of the users in the IMDb1M dataset submitted only one rated review (Section 3.3.2). However, this high percentage of users with one rating is not unique to IMDb. Another example comes from the product review domain: in Jindal and Liu's (2008) dataset of almost 6 million Amazon reviews (with ratings), about 69% of the users who

[6]The attribute-based model's stability is probably due in part to the fact that it adapts itself to the attribute values, and is therefore insensitive to small variations in the topic distributions that are due to sampling differences.

submitted reviews wrote only one review. This phenomenon is not limited to review datasets: about 58% of the users included in the full BookCrossing dataset – which contains more than 430,000 ratings – submitted only one rating (Ziegler et al., 2005). One thing that we cannot observe based on these datasets is the number of users who never submit any ratings, but this number generally includes all the potential users of the system. Therefore, our focus on such users is well-justified.

We ran our experiments on the MovieLens and IMDb1M datasets (Section 3.3), and compared the root mean square error (RMSE) on the test ratings in both cases (Section 3.2). The MovieLens dataset contains demographic information about the users, while the IMDb1M dataset contains user-generated texts (movie reviews and message board posts). We expect the RMSEs obtained on the MovieLens dataset to be lower than on the IMDb1M dataset, because MovieLens uses a rating scale of five stars while IMDb1M ratings are given on a ten-star scale.[7] Hence, the results are not comparable across datasets (and the RMSE ranges in the result plots vary according to the dataset). In addition, the user-item rating matrix of the MovieLens dataset is less sparse than the IMDb1M matrix, making rating prediction more challenging in the latter case. Nonetheless, employing these two datasets allows us to test our approach with two different sources of attributes by comparing the attribute-based model to competitive baselines.

In preliminary experiments, we found that setting $F = 10, \gamma_0 = 0.01$, and $\lambda_\gamma = 0.995$, and running Algorithms 7.1 and 7.2 for 1,000 steps yields good results. Therefore, we leave these parameter values constant in all the experiments. We avoided a bias towards users with a certain number of ratings by using cross validation to tune the regularisation parameters $\lambda_1, \ldots, \lambda_6$ independently of the GivenX value. For the MovieLens dataset, this yielded $\lambda_1 = \lambda_2 = 0.06$, $\lambda_3 = \lambda_4 = 0.01$, $\lambda_5 = 0.01$ and $\lambda_6 = 0.1$, and for the IMDb1M dataset this yielded $\lambda_1 = \lambda_2 = 1.0$, $\lambda_3 = \lambda_4 = 0.68$, $\lambda_5 = 0.1$ and $\lambda_6 = 0.01$. Hence, all the methods were run under identical conditions for all GivenX values, and the only thing that changed was the method itself. It is worth noting that in preliminary experiments we found that small changes in the values of the regularisation parameters did not yield statistically significant differences in performance.

7.4.2 MF and the Number of User Ratings

As mentioned in Section 7.2, user-based MF is expected to suffer from the new user problem. That is, it is likely to perform poorly for users with a small number of ratings. To verify this hypothesis we performed an experiment on the MovieLens and

[7] We did not make any adjustments to the rating scales because we did not want to distort the meaning of the ratings (e.g., such a distortion could occur when mapping five-star ratings to a ten-star scale, since mapping 3/5 to either 5/10 or 6/10 does not result in a rating that is the exact middle of the scale like 3/5).

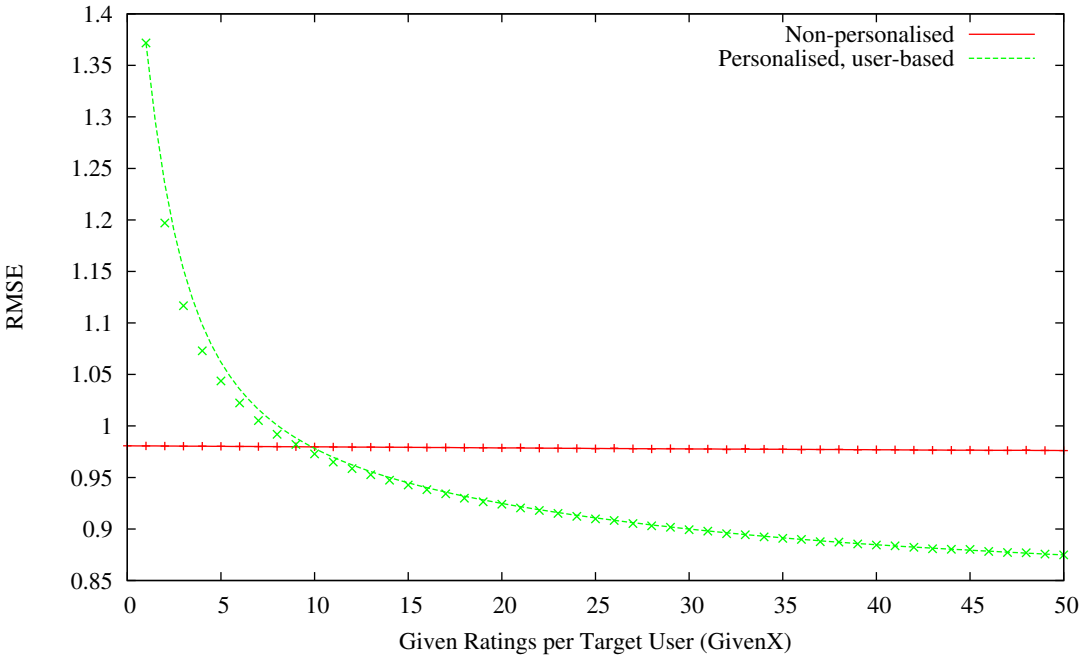


Figure 7.2: MF and the number of user ratings (Given0–50, dataset: MovieLens)

IMDb1M datasets. We employed the GivenX protocol with X values of $0, 1, \ldots, 50$, and compared the RMSE obtained when using non-personalised prediction (i.e., only the global mean and the item bias: $\mu + b_i^{(I)}$) to that obtained when using personalised, user-based prediction, as defined in Equation 7.2.

The results of this experiment are presented in Figures 7.2 (for the MovieLens dataset) and 7.3 (for the IMDb1M dataset). As the figures show, the personalised user-based model outperformed the non-personalised model when at least ten ratings were available for each user on MovieLens, but it required only two given ratings on IMDb1M.[8] In general, this means that the personalised user-based model can be used to yield accurate predictions for users who are willing to supply some explicit ratings, but is unlikely to perform as well on users that are less active.

A notable difference between the MovieLens and IMDb1M results is that for IMDb1M the performance of the non-personalised model visibly improved with the increase in GivenX values, while it improved to a lesser degree for MovieLens. This may be because the overall number of training ratings increased with the GivenX values, which caused an increase in the average number of ratings per item (we observed a similar phenomenon in the GivenX experiment results presented in Section 6.5.3). The number of ratings per item directly affects the performance of the non-personalised baseline, which relies on these ratings for the calculation of the item bias. The smaller difference between low GivenX results and high GivenX results on the MovieLens dataset is because it contains more ratings and is less

[8]The differences between the two methods for each GivenX value are statistically significant in all cases except for Given9 on the MovieLens dataset.

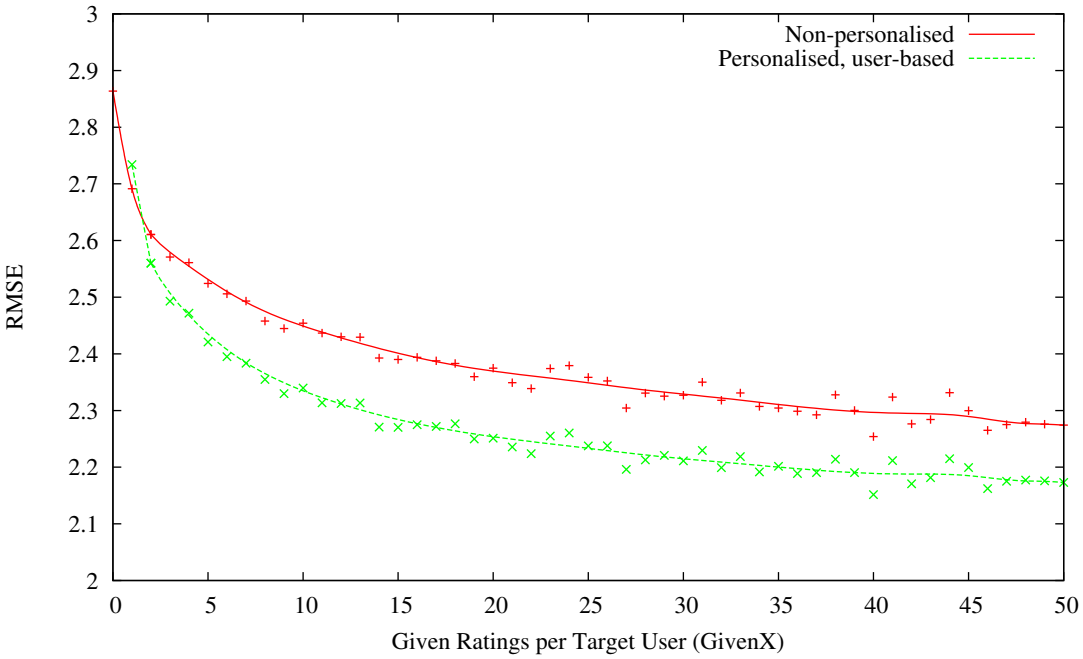


Figure 7.3: MF and the number of user ratings (Given0–50, dataset: IMDb1M)

sparse than IMDb1M (this is also the reason why the IMDb1M results do not fall on a smooth line as the MovieLens results do). Nonetheless, our focus here is on the difference between the non-personalised baseline and the user-based model for each GivenX value. These consistent differences support our hypothesis that the user-based model performs poorly for users with few ratings, and motivate the use of our attribute-based model in such cases.

This experiment can also be seen as establishing a bound on our expectations from the attribute-based model. Specifically, the absolute improvement in terms of RMSE in the Given50 case when switching from non-personalised prediction to the user-based model is about 0.101 for both MovieLens and IMDb1M (the relative improvement is 11.58% for MovieLens and 4.65% for IMDb1M). When explicit ratings are unavailable, it would be unrealistic to hope for larger improvements, unless the information we have about the users is of very high quality and of great relevance to the rating prediction task. Note that while these differences may seem small, such small differences in RMSE have been shown to yield large differences in the quality of top-N recommendation lists (Koren, 2008). Moreover, when considering the end goal of the rating prediction algorithms, which is to serve as a basis for recommendations, an algorithm that yields personalised predictions may be preferable to a non-personalised algorithm, even if their RMSEs are comparable, since the use of non-personalised predictions is likely to result in recommendations of the most popular items, which users already know about.

Method	Given0	Given1
Non-personalised	0.981	0.981
Personalised, user-based	—	1.372
Personalised, attribute-based	**0.976**	**0.976**

Table 7.1: MFUA with demographic attributes (Given0 & Given1, dataset: MovieLens)

7.4.3 MFUA with Demographic Attributes

In this section, we evaluate the performance of MFUA (Section 7.2) on the MovieLens dataset (Section 3.3.6). This dataset includes some demographic information about the users: age, gender, occupation and postcode. We followed the process described in Section 7.3.2 to convert this demographic information into 85 probabilistic attributes. In all the experiments, we compared the RMSE obtained with our attribute-based model to the RMSEs obtained with the user-based model and the non-personalised model (Section 7.1).

In our first experiment, we ran MFUA on the full set of attributes under the Given0 and Given1 protocols. The results are presented in Table 7.1 (the best statistically significant results for each GivenX value are highlighted in boldface).[9] As we can see, our attribute-based approach obtained the same absolute improvement over the non-personalised baseline in both cases. While the improvement over the non-personalised baseline is small, the fact that even some improvement was obtained is encouraging, as it indicates that recommender systems based on our attribute-based approach may offer a certain degree of personalisation for new users without compromising the expected quality of the predictions. Moreover, the more substantial difference between the attribute-based and user-based results in the Given1 case suggests that it would be prudent to generate personalised predictions for new users using the attribute-based model rather than with the user-based model.

In our second experiment, we employed PCA to transform the original attributes to attributes that account for most of the variability in the demographic data (Section 7.3.2). We then ran MFUA with the transformed attributes under the Given0 and Given1 scenarios. As in the first experiment, the Given0 results were similar to the Given1 results. In both cases, comparable performance to using the full set of attributes was obtained when three principal components were retained (RMSE of 0.976).[10] Using less principal components yielded worse performance (RMSEs of 0.978 and 0.977 with one and two principal components respectively), while using

[9] The baseline results are the same as the Given0 and Given1 results from Figure 7.2.

[10] The overall number of attributes in this case was 12 since we split the attribute values into quartiles (Section 7.3.2).

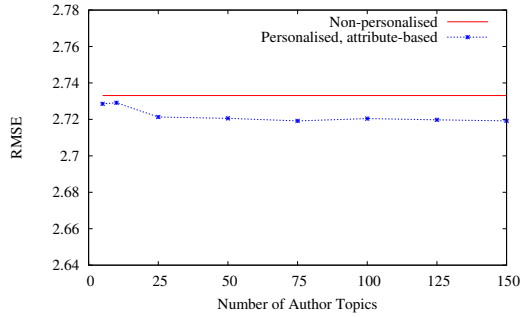
more yielded performance comparable to using three principal components (we experimented with up to ten principal components). It is unsurprising that employing PCA did not improve performance over using the full set of attributes, because the original attributes only account for four different user characteristics. Hence, it is unlikely that overfitting was a problem in this case. Nonetheless, the fact that using the transformed attributes yielded comparable performance to using the full set of attributes indicates that employing PCA is an effective way of dealing with a large number of attributes. Unfortunately, we did not have access to a dataset with richer user information, so testing this conjecture is left for future work.

7.4.4 MFUA with Text-based Attributes

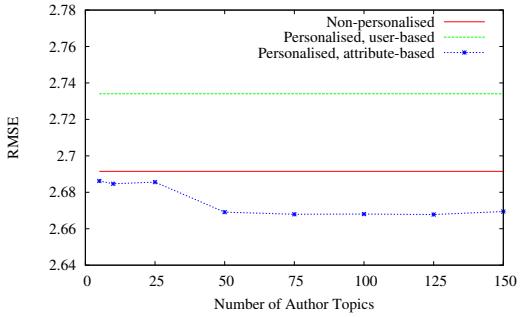
The results obtained with *explicit* demographic information indicate that using our MFUA model to consider user information can yield accurate and personalised rating predictions for new users. However, our main goal in this chapter is to show that user texts can be effectively used as a source of *implicit* information that can be harnessed to improve the accuracy of rating predictions for new users. Hence, in this section, we consider the common case where there are no explicit user demographics available, but some texts by the users are given. In this case, we represent users by distributions over author topics inferred from their texts using either AT or DADT (Section 7.3.3). We test our approach on the IMDb1M dataset, focusing on users who submitted no explicit ratings but have written some message board posts, and users who submitted exactly one rated review. We could not use the MovieLens dataset for the experiments in this section, because it does not contain any user-generated texts. Hence, the results presented in this section are not directly comparable to the MovieLens results presented in the previous section.

Attributes Inferred Using the AT Model

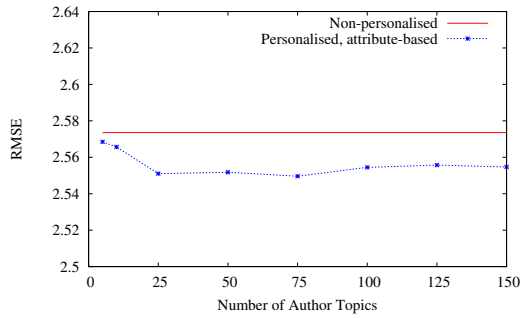
As discussed in Section 7.2, MFUA is expected to be sensitive to the number of attributes (in this case, the number of author topics). Hence, we first ran an experiment using the AT model where we varied the number of topics from 5 to 150, used the default prior values from Section 4.1.2, and retained the 1000th sample from the Gibbs sampling chain. We employed the GivenX protocol with X values of 0 and 1, as done in the MovieLens experiments. In the Given0 case, we only report results based on users who have written at least one message board post, because no attributes can be inferred for users who submitted no posts. In the Given1 case, each user has at least one rating with its corresponding review in the training set, and thus we could infer attributes for all the target users (i.e., we included users with and without posts in the test set in this case). In addition, we present the results obtained for users with at least ten texts each (ten posts in the Given0 case and nine posts and one review in the Given1 case), since user representations in this



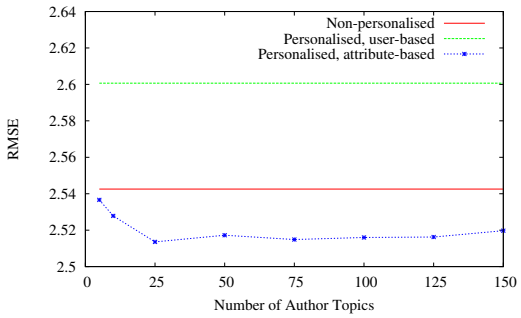
(a) Given0 (at least one text per user)



(b) Given1 (at least one text per user)



(c) Given0 (at least ten texts per user)



(d) Given1 (at least ten texts per user)

Figure 7.4: MFUA with AT attributes (Given0 & Given1, dataset: IMDb1M)

case are expected to be more meaningful than for users with only one text (either a review or a post). In all these cases, *training* users with no texts were included in the training set, and the model was not retrained to account for target users with a specific number of texts.

Figure 7.4 presents the results of this experiment. The Given0 results are shown in the left plots and the Given1 results are in the right plots; results for test sets where the users were restricted to have at least one text each appear in the top plots, and results where the per-user text threshold was ten are presented in the bottom plots.[11] Since the test set populations are different in each of the four cases, the most meaningful comparisons are between methods within the same test set. Nonetheless, it is worth noting that the lower Given1 RMSEs in comparison to Given0 are consistent with the results obtained in Section 7.4.2. In addition, the lower RMSEs obtained on target users with at least ten texts each in comparison to users with at least one text each (by all the methods, even those that do not rely on texts) may be attributed to the more prolific users being more likely to be genuine users that are easier to model than, e.g., users that signed up only to post a single review on a movie that they feel strongly about. However, it is important to remember that the IMDb1M dataset contains three times more posts than reviews, and that according to our random sample of IMDb users, the ratio of users who

[11]Note that all the plots use the same y-scale, but with different RMSE ranges: [2.64, 2.78] for the top plots and [2.50, 2.64] for the bottom plots.

participate as message posters to users who participate as reviewers is about five to three (Section 3.3.2). Hence, we believe that modelling users non-intrusively based on posts is potentially of benefit to a large number of users. In addition, our positive results on posts indicate that our approach may be applicable to other types of text, such as emails and social media messages, which are available in many realistic scenarios (assuming user consent is given to use these texts).

As Figure 7.4 shows, our attribute-based approach did not require many topics to outperform the baselines.[12] In the Given0 scenarios, the best performance was obtained with 75 topics, but using 25 topics yielded comparable results.[13] In the Given1 scenarios, the best performance was obtained with 125 topics when at least one text per target user was available (with 50 topics yielding comparable results), and 25 topics with at least ten texts per target user.[14] An interesting point to note is that the best relative improvement in RMSE over the non-personalised baseline was higher for target users with ten texts than for target users with only one text (0.94% compared to 0.51% in the Given0 case and 1.05% compared to 0.89% in the Given1 case). This result is unsurprising, and it supports our hypothesis that considering user texts with topical user models in the MFUA model can yield personalised rating predictions that are more accurate than the baselines. Moreover, these relative improvements are considerable in comparison to the approximate bound of 4.65% that we established in Section 7.4.2, and are higher than the relative improvement obtained in the MovieLens experiments (0.51%). While IMDb1M results and MovieLens results are not directly comparable, the reason for the difference in relative improvements may be that the texts we considered carry more rating-related information about users and their interests than the demographic information included in the MovieLens dataset. Further experiments on a rating dataset that contains both demographic information and user texts are needed to verify this assumption. We leave such experiments for future work, as we do not have access to such a dataset.

Attributes Inferred Using the DADT Model

Our second set of experiments explored the use of DADT as a source of text-based attributes. Since the overall number of attributes (i.e., author topics) is expected

[12]The differences between the attribute-based model and the baselines are statistically significant in all cases except for Given0 with 10 topics and Given1 with 25 topics, both with at least one text per target user.

[13]The differences between the best Given0 results (with 75 topics) and results obtained with other topic numbers are statistically significant in all cases except for 25, 50, 100, 125 and 150 topics for target users with at least one text each, and 25 and 50 topics for target users with at least ten texts each.

[14]The differences between the best Given1 results (with 125 topics for target users with at least one text each, and 25 topics for target users with at least ten texts each) and results obtained with other topic numbers are statistically significant in all cases except for 50, 75, 100 and 150 topics for target users with at least one text each, and 50, 75, 100 and 125 topics for target users with at least ten texts each.

Method	Given0		Given1	
Texts/User:	≥ 1	≥ 10	≥ 1	≥ 10
Non-personalised	2.733	2.574	2.691	2.543
Personalised, user-based	—	—	2.734	2.601
Personalised, AT-based	**2.719**	**2.550**	**2.668**	**2.514**
Personalised, DADT-based				
$T^{(D)} = 1$	**2.720**	2.555	2.674	**2.517**
$T^{(D)} = 5$	**2.719**	**2.551**	2.678	**2.514**
$T^{(D)} = 10$	**2.721**	2.555	2.681	**2.519**
$T^{(D)} = 25$	**2.724**	2.561	2.681	**2.517**

Table 7.2: MFUA with DADT attributes (Given0 & Given1, dataset: IMDb1M)

to have a strong effect on MFUA's performance, we compared the performance obtained with AT to DADT by varying the number of *document* topics $T^{(D)}$ from 1 to 25, while leaving the number of author topics constant at 75 for Given0 and 125 for Given1, as these yielded the best results for AT on target users with at least one text each (Figure 7.4). We used the default prior values from Section 4.1.2 together with the prior values that yielded the best DADT performance in our authorship attribution experiments on IMDb1M (Section 5.3.4), and retained the 1000th sample from the Gibbs sampling chain, as done in the AT experiment.

Table 7.2 shows the results of this experiment. As for the AT experiments, we present the Given0 and Given1 results obtained for target users with at least one text or at least ten texts each. As we can see, DADT attributes did not yield better performance than AT attributes, and employing DADT in the Given1 case yielded worse performance than using AT attributes for target users with at least one text. Unsurprisingly, using only a few document topics $T^{(D)}$ yielded the best results of the DADT variants, because DADT with $T^{(D)} = 0$ is equivalent to AT. The reason DADT did not outperform AT may be that DADT tends to yield user representations that help discriminate between texts by individual users (as shown in our authorship attribution experiments in Chapter 5), but such representations are not as useful when employed as attributes for MFUA, since the attribute-based model requires a soft clustering of the users (Section 7.3).

7.5 Summary and Conclusions

In this chapter, we introduced the *Matrix Factorisation with User Attributes* (MFUA) model, which considers user attributes when generating rating predictions for new users. We evaluated our MFUA model on explicit demographic attributes, and on

implicit attributes inferred from user-generated texts via topical user models (Chapter 4). As we showed in Section 7.4, using MFUA to consider either type of user attributes in rating prediction yields improved accuracy over traditional matrix factorisation (MF) and over a non-personalised baseline for users who submitted few or no ratings. These results demonstrate the usefulness of our topical user modelling approach and indicate that considering texts when generating rating predictions can potentially improve the quality of recommendations for new users.

While the improvements we obtained may seem small, we must remember that small improvements in predictive accuracy may yield large improvements in recommendation quality (Koren, 2008). In addition, we focused on new users who submitted no ratings or only one explicit rating – a population segment that often forms the majority of the user population, but for which rating prediction is known to be challenging. Personalising the experience of such users may encourage them to become more active. Hence, improved rating predictions for these users can eventually make the rating prediction task easier as users get more involved and supply more ratings.

Nonetheless, an important question that we could not definitively answer under our experimental setup is whether users would find recommendations based on our MFUA model better than recommendations based on the baselines. While offline studies that focus on improving accuracy provide important insights into the performance of rating prediction algorithms, accuracy measures do have their limitations (McNee et al., 2006). For example, the errors of two methods on some items might be completely different, but they may cancel each other out and thus have the same RMSE overall, so the differences between the methods would not be reflected by the RMSE. To the best of our knowledge, only an online study can reliably compare competing recommendation algorithms and their underlying rating prediction methods, but to perform such a study one would need to implement a real system that users want to use. Therefore, an interesting future work direction is to deploy and test our MFUA model as part of such real systems.

Chapter 8

Conclusion

Recent years have seen an abundance of user-generated texts published online. This abundance creates opportunities and poses challenges in the growing research areas of text mining and user modelling. This thesis investigated two aspects of these research areas: the extraction of implicit information from user-generated texts to create user models, and the application of these models to tasks that require user information. One of our main focus areas was on representing users as distributions over topics via topical user models. We employed these topical user models in methods we developed to address the following tasks: (1) authorship attribution: identifying which user wrote a given anonymous text; (2) polarity inference: detecting the level of sentiment expressed in a given text; and (3) rating prediction: determining a given user's expected sentiment towards a given item. This chapter summarises the contributions of this thesis (Section 8.1), presents directions for future work (Section 8.2), and concludes the thesis (Section 8.3).

8.1 Summary of Contributions

In Chapter 4, we introduced the notion of *topical user models* – using topic models to obtain compact representations of users based on their texts. These topical user models aim to capture the interests of authors together with aspects of their authorship style, which is indicative of characteristics such as demographic information and personality traits. We discussed how two previously-suggested topic models, *Latent Dirichlet Allocation* (LDA) and the *Author-Topic* (AT) model, can be used to represent users. In addition, we presented our *Disjoint Author-Document Topic* (DADT) model, which combines LDA and AT into a single model in order to separate information about the authors from information about the documents. We demonstrated the merits of our DADT model in comparison to LDA and AT through experiments on a synthetic dataset, and presented the results of a preliminary study on considering word order in topical user models. In general, topical user models may potentially be used for any task that can benefit from user information

extracted from texts. In subsequent chapters, we demonstrated the applicability of topical user models to the three tasks considered in this thesis: authorship attribution, polarity inference and rating prediction.

In Chapter 5, we evaluated the topic models from Chapter 4 in the context of the *authorship attribution* task. We tested several approaches, and showed that methods based on our DADT model yield state-of-the-art performance in several scenarios where the number of candidate authors ranges from three to about 20,000. These results provide empirical evidence that topical user models retain information from user-generated texts that is representative of user characteristics (in addition to the established finding that such models capture user interests). In addition, our results lay the foundation for further research into the application of topic modelling techniques to authorship analysis, as outlined in Section 8.2.

In Chapter 6, we addressed the *polarity inference* task in a user-aware manner by developing a framework that combines the outputs of user-specific inference models. We showed that even when all the models are given equal weights, our approach outperforms two baselines: one that ignores any information about the users, and another that employs only the identity of the author of the text without considering texts by other users. In addition, we showed that weighting the user-specific models based on user rating patterns and language use (e.g., as captured by topical user models) can improve results even further. These results support our hypothesis that the way sentiment is expressed is often author-dependent, and show that our approach successfully harnesses this dependency to improve polarity inference performance.

In Chapter 7, we moved from inferring user sentiments to predicting them. We introduced a *rating prediction* framework that considers user attributes when generating rating predictions for new users with a matrix factorisation algorithm. We showed that improvements in rating prediction performance are obtained by considering either demographic attributes or attributes inferred from texts using topical user models. While previous studies have demonstrated that demographic information can be used to improve predictive accuracy (though not within the matrix factorisation framework), the main contribution of Chapter 7 is in showing that general texts by new users, such as message board posts, can be employed to yield personalised predictions that are more accurate than predictions yielded by baselines that consider only ratings.

8.2 Future Work Directions

The work presented in this thesis can be extended in many ways. This section discusses the main future work directions that arise from each of the chapters.

In Chapter 4, we discussed possible approaches to considering word order in author-aware topic models. While preliminary testing of these approaches on the authorship attribution task yielded unsatisfactory results, it is possible that further work on integrating word order into author-aware topic models would yield better results. Specifically, as mentioned in Section 4.5.2, extending our HMM-LDA-AT model to reduce the number of configurable parameters may allow it to obtain better authorship attribution results. In addition, performance may also be improved by extending the author-aware topic models from Chapter 4 to consider additional feature types (e.g., part-of-speech tags). A related direction is to explore the topical user models yielded by generic topic models that support arbitrary feature types (as discussed in Section 2.2.2, generic models may be used to generate author representations by defining a metadata label for each author). Our models and the extended models may potentially be applied to any scenario where user texts are available, going beyond the three tasks considered in this thesis.

Extensions to our authorship attribution work (Chapter 5) include automatically inferring the document/author topic balance in our DADT model, employing DADT in semi-supervised classification, and addressing the open-set authorship attribution task (where the authors of some test texts may not be in the training set of candidate authors). The former two require changes to DADT's training procedure, while the open-set task can be addressed by learning a threshold on the probability of the chosen author (returning "unknown author" if this probability is less than the threshold). In addition, as discussed in Section 5.3.4, a DADT-SVM ensemble may potentially yield substantial performance improvements. It would also be interesting to develop topic models that account for differences between authors' writing styles that stem from characteristics such as age and gender, as opposed to differences at the individual level. Accounting for such differences would enable the use of topic models in authorship profiling tasks, where the goal is to infer author characteristics from texts. Finally, running our reviewer identification experiment (Section 5.4) on a larger corpus would allow the research community to gauge whether anonymous reviewers are indeed anonymous.

Our study of user-aware polarity inference (Chapter 6) opens the door to further research on how improved sentiment analysis results can be obtained by considering authorship information. As our focus was on demonstrating the benefits of taking users into account, we used generic inference models as the base inferrers in the experiments with our polarity inference framework. Improved results may be obtained by employing base inferrers that are specifically designed for polarity inference, or by extending existing sentiment analysis methods to consider authorship information. An additional research direction is complementary to the work presented in Chapter 6 – considering textual polarity when performing authorship attribution

of sentiment-bearing texts (as opposed to harnessing authorship information when performing polarity inference). We conjecture that accounting for differences in language use between positive and negative texts may yield improved authorship attribution results, due to the connection between authors' identity and the way they express their sentiments, which we established in Chapter 6.

An obvious future work direction on our rating prediction framework (Chapter 7) is to evaluate the quality of recommendations generated based on our rating prediction methods. This should be done as part of a live system, which can potentially access texts of different types, going beyond the reviews and message board posts we considered in our experiments.[1] Another direction involves using sentiment analysis techniques to obtain user attributes that can be employed within our rating prediction framework. For example, Lin and He's (2009) sentiment-aware extensions to LDA could be used to extract sentiments towards concepts (from general texts – not only reviews), which may be integrated into our framework as user attributes.

8.3 Concluding Remarks

This thesis focused on the extraction of implicit information from user-generated texts to create user models, and the application of these models to three tasks that require user information. As we have shown, drawing on previous research in several fields allowed us to obtain improved performance on the tasks we considered:

- Employing topical user models (specifically, our DADT model) in authorship attribution yielded state-of-the-art performance in several scenarios where the number of candidate authors ranges from three to about 20,000.
- Harnessing authorship information allowed us to obtain polarity inference results that are better than those obtained by baselines that either ignore authors or consider texts only by a single author.
- Employing topical user models inferred from texts by new users (with a few or no ratings) made it possible to generate personalised rating predictions for such users that are more accurate than predictions based only on ratings.

While the work done in this thesis advances the state of the art on the tasks we considered, there is still much more that could be done (Section 8.2). We hope that this thesis will help inspire future research that would further explore the connections between the fields we considered, possibly integrating more areas while addressing related tasks. As demonstrated throughout this thesis by our empirical results, exploring and utilising these connections may yield improved performance on a variety of tasks.

[1] User consent should be explicitly obtained before employing private texts such as emails and social media messages.

Bibliography

Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham. Informed recommender agent: Utilizing consumer product reviews through text mining. In *IADM 2006: Proceedings of the International Workshop on Interaction between Agents and Data Mining*, pages 37–40, Hong Kong, 2006.

Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham. Informed recommender: Basing recommendations on consumer product reviews. *IEEE Intelligent Systems*, 22(3):39–47, 2007.

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 217–253. Springer US, 2011.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD 1993: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, DC, USA, 1993.

Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, and Nuria Oliver. The wisdom of the few: A collaborative filtering approach based on expert opinions from the web. In *SIGIR 2009: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 532–539, Boston, MA, USA, 2009.

Sarabjot Singh Anand, Dietmar Jannach, Bamshad Mobasher, and Alfred Kobsa, editors. *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems*, Barcelona, Spain, 2011.

Shlomo Argamon and Patrick Juola. Overview of the international authorship identification competition at PAN-2011. In *CLEF 2011: Proceedings of the 2011*

Conference on Multilingual and Multimodal Information Access Evaluation (Lab and Workshop Notebook Papers), Amsterdam, The Netherlands, 2011.

Shlomo Argamon, Sushant Dhawle, Moshe Koppel, and James W. Pennebaker. Lexical predictors of personality type. In *Proceedings of the Joint Annual Meeting of the Interface and the Classification Society of North America*, St. Louis, MO, USA, 2005.

Shlomo Argamon, Moshe Koppel, James W. Pennebaker, and Jonathan Schler. Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119–123, 2009.

Marko Balabanovic and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

Jayanta Basak, Anant Sudarshan, Deepak Trivedi, and M. S. Santhanam. Weather data mining using independent component analysis. *Journal of Machine Learning Research*, 5(Dec):239–253, 2004.

Brigitte Bigi, Armelle Brun, Jean-Paul Haton, Kamel Smaïli, and Imed Zitouni. A comparative study of topic identification on newspaper and e-mail. In *SPIRE 2001: Proceedings of the 8th International Symposium on String Processing and Information Retrieval*, pages 238–241, Laguna de San Rafael, Chile, 2001.

David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4): 77–84, 2012.

David M. Blei and John D. Lafferty. Topic models. In Ashok Srivastava and Mehran Sahami, editors, *Text Mining: Classification, Clustering, and Applications*, pages 71–93. Chapman and Hall/CRC, 2009.

David M. Blei and Jon D. McAuliffe. Supervised topic models. In *NIPS 2007: Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, pages 121–128, Vancouver, BC, Canada, 2007.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. In *NIPS 2001: Proceedings of the 15th Annual Conference on Neural Information Processing Systems*, pages 601–608, Vancouver, BC, Canada, 2001.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

Fabian Bohnert, Daniel F. Schmidt, and Ingrid Zukerman. Spatial processes for recommender systems. In *IJCAI 2009: Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 2022–2027, Pasadena, CA, USA, 2009.

Jordan Boyd-Graber, David Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *EMNLP-CoNLL 2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1024–1033, Prague, Czech Republic, 2007.

John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI 1998: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, WI, USA, 1998.

Michael Brennan and Rachel Greenstadt. Practical attacks against authorship recognition techniques. In *IAAI 2009: Proceedings of the 21st Conference on Innovative Applications of Artificial Intelligence*, pages 60–65, Pasadena, CA, USA, 2009.

Armelle Brun, Sylvain Castagnos, and Anne Boyer. A positively directed mutual information measure for collaborative filtering. In *SIIE 2009: Proceedings of the 2nd International Conference on Information Systems and Economic Intelligence*, pages 943–958, Hammamet, Tunisia, 2009.

Armelle Brun, Sylvain Castagnos, and Anne Boyer. Social recommendations: Mentor and leader detection to alleviate the cold-start problem in collaborative filtering. In I-Hsien Ting, Tzung-Pei Hong, and Leon S. L. Wang, editors, *Social Network Mining, Analysis and Research Trends: Techniques and Applications*, pages 270–290. IGI Global, 2011.

Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

Iván Cantador, Peter Mika, David Vallet, José C. Cortizo, and Francisco M. Carrero, editors. *Proceedings of the 1st International Workshop on Adaptation, Personalization and Recommendation in the Social-semantic Web*, Heraklion, Greece, 2010.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS 2009: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 288–296, Vancouver, BC, Canada, 2009.

Eugene Charniak. Statistical techniques for natural language parsing. *AI Magazine*, 18(4):33–44, 1997.

Carole E. Chaski. Who's at the keyboard? Authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, 4(1), 2005.

Sajib Dasgupta and Vincent Ng. Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In *ACL-IJCNLP 2009: Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 701–709, Singapore, 2009.

Hal Daumé III. De-authorship attribution, 2012. Retrieved from `nlpers.blogspot.com.au/2012/02/de-authorship-attribution.html` on 20 March, 2012 (personal blog post).

Scott Deerwester, Susan Dumais, Thomas Landauer, George Furnas, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

Marco Degemmis, Ernesto William De Luca, Tommaso Di Noia, Aldo Gangemi, Michael Hausenblas, Pasquale Lops, Thomas Lukasiewicz, Till Plumbaum, and Giovanni Semeraro, editors. *Proceedings of the 2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation*, Bonn, Germany, 2011.

Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.

Gideon Dror, Noam Koenigstein, and Yehuda Koren. Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *RecSys 2011: Proceedings of the 5th ACM Conference on Recommender Systems*, pages 165–172, Chicago, IL, USA, 2011.

Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101 (Suppl. 1):5220–5227, 2004.

Sandra Garcia Esparza, Michael P. O'Mahony, and Barry Smyth. A multi-criteria evaluation of a user-generated content based recommender system. In *RSWEB 2011: Proceedings of the 3rd ACM RecSys Workshop on Recommender Systems and the Social Web*, Chicago, IL, USA, 2011.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9(Aug):1871–1874, 2008.

Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. Developing constraint-based recommenders. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 187–215. Springer US, 2011.

Agner Fog. Calculation methods for Wallenius' noncentral hypergeometric distribution. *Communications in Statistics, Simulation and Computation*, 37(2):258–273, 2008.

Michael Gamon. Linguistic correlates of style: Authorship classification with deep linguistic analysis features. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 611–617, Geneva, Switzerland, 2004a.

Michael Gamon. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 841–847, Geneva, Switzerland, 2004b.

Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM 2010: Proceedings of the 10th IEEE International Conference on Data Mining*, pages 176–185, Sydney, NSW, Australia, 2010.

Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB 2009: Proceedings of the 12th International Workshop on the Web and Databases*, Providence, RI, USA, 2009.

Gayatree Ganu, Yogesh Kakodkar, and Amélie Marian. Improving the quality of predictions using textual information in online user reviews. *Information Systems*, 2012. Advance access published online March 13, 2012 doi:10.1016/j.is.2012.03.001.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

SongJie Gong. Employing user attribute and item attribute to enhance the collaborative filtering recommendation. *Journal of Software*, 4(8):883–890, 2009.

Stephan Greene and Philip Resnik. More than words: Syntactic packaging and implicit sentiment. In *NAACL-HLT 2009: Proceedings of the Human Language*

Technology Conference of the North American Chapter of the Association for Computational Linguistics, pages 503–511, Boulder, CO, USA, 2009.

Thomas L. Griffiths and Mark Steyvers. A probabilistic approach to semantic representation. In *CogSci 2002: Proceedings of the 24th Annual Conference of the Cognitive Science Society*, pages 381–386, Fairfax, VA, USA, 2002a.

Thomas L. Griffiths and Mark Steyvers. Prediction and semantic association. In *NIPS 2002: Proceedings of the 16th Annual Conference on Neural Information Processing Systems*, pages 11–18, Vancouver, BC, Canada, 2002b.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, 2004.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. Integrating topics and syntax. In *NIPS 2004: Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, pages 537–544, Vancouver, BC, Canada, 2004.

Trish Groves. Is open peer review the fairest system? Yes. *BMJ*, 341:c6424, 2010.

Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic Markov models. In *AISTATS 2007: Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 163–170, San Juan, Puerto Rico, 2007.

Weiwei Guo and Mona Diab. Semantic topic models: Combining word distributional statistics and dictionary definitions. In *EMNLP 2011: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 552–561, Edinburgh, UK, 2011.

Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3): 203–259, 2001.

Morgan Harvey, Mark J. Carman, Ian Ruthven, and Fabio Crestani. Bayesian latent variable models for collaborative item rating prediction. In *CIKM 2011: Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 699–708, Glasgow, UK, 2011.

Marti A. Hearst. Untangling text data mining. In *ACL 1999: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 3–10, College Park, MD, USA, 1999.

Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR 1999: Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, Berkeley, CA, USA, 1999.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

Thomas Hofmann. Probabilistic latent semantic analysis. In *UAI 1999: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 289–296, Stockholm, Sweden, 1999.

Thomas Hofmann. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In *SIGIR 2003: Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 259–266, Toronto, ON, Canada, 2003.

Timothy Hospedales, Shaogang Gong, and Tao Xiang. A Markov clustering topic model for mining behaviour in video. In *ICCV 2009: Proceedings of the IEEE 12th International Conference on Computer Vision*, pages 1165–1172, Kyoto, Japan, 2009.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2003.

Rong Hu and Pearl Pu. Enhancing collaborative filtering systems with personality information. In *RecSys 2011: Proceedings of the 5th ACM Conference on Recommender Systems*, pages 197–204, Chicago, Illinois, USA, 2011.

Francisco Iacobelli, Alastair Gill, Scott Nowson, and Jon Oberlander. Large scale personality classification of bloggers. In *MLAC 2011: Proceedings of the International Workshop on Machine Learning for Affective Computing*, pages 568–577, Memphis, TN, USA, 2011.

Peter Jackson and Isabelle Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. John Benjamins Publishing Company, 2nd edition, 2007.

Michael Jahrer, Andreas Töscher, and Robert Legenstein. Combining predictions for accurate recommender systems. In *KDD 2010: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 693–702, Washington, DC, USA, 2010.

Niklas Jakob, Stefan Hagen Weber, Mark-Cristoph Müller, and Iryna Gurevych. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *TSA 2009: Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement*, pages 57–64, Hong Kong, 2009.

Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys 2010: Proceedings of the 4th ACM Conference on Recommender Systems*, pages 135–142, Barcelona, Spain, 2010.

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, 2010.

Nitin Jindal and Bing Liu. Opinion spam and analysis. In *WSDM 2008: Proceedings of the 1st International Conference on Web Search and Web Data Mining*, pages 219–230, Palo Alto, CA, USA, 2008.

Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML 1998: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, Germany, 1998.

Ian T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2nd edition, 2002.

Patrick Juola. Ad-hoc authorship attribution competition. In *ALLC-ACH 2004: Proceedings of the 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities*, pages 175–176, Göteborg, Sweden, 2004.

Patrick Juola. Authorship attribution. *Foundations and Trends in Information Retrieval*, 1(3):233–334, 2006.

Gary Kacmarcik and Michael Gamon. Obfuscating document stylometry to preserve author anonymity. In *COLING-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Main Conference Poster Sessions)*, pages 444–451, Sydney, NSW, Australia, 2006.

Judy Kay and Bob Kummerfeld. Lifelong user modelling goals, issues and challenges. In *Proceedings of the UMAP 2009 Lifelong User Modelling Workshop*, pages 27–34, Trento, Italy, 2009.

Roman Kern, Christin Seifert, Mario Zechner, and Michael Granitzer. Vote/veto meta-classifier for authorship identification. In *CLEF 2011: Proceedings of the*

2011 Conference on Multilingual and Multimodal Information Access Evaluation (Lab and Workshop Notebook Papers), Amsterdam, The Netherlands, 2011.

Alfred Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1):49–63, 2001.

Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

Moshe Koppel and Jonathan Schler. Authorship verification as a one-class classification problem. In *ICML 2004: Proceedings of the 21st International Conference on Machine Learning*, pages 62–68, Banff, AB, Canada, 2004.

Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17 (4):401–412, 2003.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. Determining an author's native language by mining a text for errors. In *KDD 2005: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 624–628, Chicago, IL, USA, 2005.

Moshe Koppel, Navot Akiva, and Ido Dagan. Feature instability as a criterion for selecting potential style markers. *Journal of the American Society for Information Science and Technology*, 57(11):1519–1525, 2006.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26, 2009.

Moshe Koppel, Navot Akiva, Idan Dershowitz, and Nachum Dershowitz. Unsupervised decomposition of a document into authorial components. In *ACL-HLT 2011: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1356–1364, Portland, OR, USA, 2011a.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94, 2011b.

Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD 2008: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434, Las Vegas, NV, USA, 2008.

Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

Yehuda Koren and Robert Bell. Advances in collaborative filtering. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer US, 2011.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

Ioannis Kourtis and Efstathios Stamatatos. Author identification using semi-supervised learning. In *CLEF 2011: Proceedings of the 2011 Conference on Multilingual and Multimodal Information Access Evaluation (Lab and Workshop Notebook Papers)*, Amsterdam, The Netherlands, 2011.

Bruce Krulwich. Lifestyle Finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.

Nidhi Kulkarni and Mark Alan Finlayson. jMWE: A Java toolkit for detecting multi-word expressions. In *MWE 2011: Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 122–124, Portland, OR, USA, 2011.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS 2008: Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, pages 897–904, Vancouver, BC, Canada, 2008.

George Lekakos and George M. Giaglis. A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. *User Modeling and User-Adapted Interaction*, 17(1):5–40, 2007.

Cane Wing-ki Leung, Stephen Chi-fai Chan, and Fu-lai Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of the ECAI Workshop on Recommender Systems*, pages 62–66, Riva del Garda, Italy, 2006.

Fangtao Li, Nathan Nan Liu, Jin Hongwei, Kai Zhao, Qiang Yang, and Xiaoyan Zhu. Incorporating reviewer and product information for review rating prediction. In *IJCAI 2011: Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1820–1825, Barcelona, Spain, 2011.

Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *CIKM 2009: Proceedings of the 18th ACM International Conference on Information and Knowledge Management*, pages 375–384, Hong Kong, 2009.

Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. Which side are you on? Identifying perspectives at the document and sentence levels. In *CoNLL 2006: Proceedings of the 10th International Conference on Computational Natural Language Learning*, pages 109–116, New York, NY, USA, 2006.

Bing Liu. Sentiment analysis and subjectivity. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, pages 627–666. Chapman and Hall/CRC, 2010.

Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463. Springer US, 2012.

Adam Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3): 8:1–8:49, 2008.

Pasquale Lops, Marco Degemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011.

Bin Lu, Chenhao Tan, Claire Cardie, and Benjamin K. Tsou. Joint bilingual sentiment classification with unlabeled parallel corpora. In *ACL-HLT 2011: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 320–330, Portland, OR, USA, 2011a.

Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA. *Information Retrieval*, 14(2):178–203, 2011b.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Yi Mao and Guy Lebanon. Isotonic conditional random fields and local sentiment flow. In *NIPS 2006: Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, pages 961–968, Vancouver, BC, Canada, 2006.

Yi Mao and Guy Lebanon. Generalized isotonic conditional random fields. *Machine Learning*, 77(2):225–248, 2009.

Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI 2006: Extended Abstracts of the 24th Conference on Human Factors in Computing Systems*, pages 1097–1101, Montreal, QC, Canada, 2006.

Thomas C. Mendenhall. The characteristic curves of composition. *Science*, 9(214S): 237–246, 1887.

David Mimno and David Blei. Bayesian checking for topic models. In *EMNLP 2011: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 227–237, Edinburgh, UK, 2011.

David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *KDD 2007: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 500–509, San Jose, CA, USA, 2007.

David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *UAI 2008: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 411–418, Helsinki, Finland, 2008.

Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *KDD 2002: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 341–349, Edmonton, AB, Canada, 2002.

Yashar Moshfeghi, Benjamin Piwowarski, and Joemon M. Jose. Handling data sparsity in collaborative filtering using emotion and semantic based features. In *SIGIR 2011: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 625–634, Beijing, China, 2011.

Frederick Mosteller and David L. Wallace. *Inference and Disputed Authorship: The Federalist.* Addison-Wesley, 1964.

Mihir Nanavati, Nathan Taylor, William Aiello, and Andrew Warfield. Herbert West – deanonymizer. In *HotSec'11: Proceedings of the 6th USENIX Workshop on Hot Topics in Security*, San Francisco, CA, USA, 2011.

David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. Statistical entity-topic models. In *KDD 2006: Proceedings of the 12th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, pages 680–686, Philadelphia, PA, USA, 2006.

Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *NIPS 2001: Proceedings of the 15th Annual Conference on Neural Information Processing Systems*, pages 841–848, Vancouver, BC, Canada, 2001.

Jon Oberlander and Scott Nowson. Whose thumb is it anyway? Classifying author personality from weblog text. In *COLING-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 627–634, Sydney, NSW, Australia, 2006.

Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman. *Mahout in Action*. Manning Publications Co., Shelter Island, NY, USA, 2011.

Georgios Paltoglou and Mike Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *ACL 2010: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1386–1395, Uppsala, Sweden, 2010.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL 2004: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 271–278, Barcelona, Spain, 2004.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL 2005: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 115–124, Ann Arbor, MI, USA, 2005.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *EMNLP 2002: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, PA, USA, 2002.

Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5–6):393–408, 1999.

Lisa Pearl and Mark Steyvers. Detecting authorship deception: A supervised machine learning approach using author writeprints. *Literary and Linguistic Computing*, 27(2):183–196, 2012.

Lawrence R. Rabiner and Biing-Hwang Juang. An introduction to hidden Markov models. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 3(1):4–16, 1986.

Arun Rajkumar, Saradha Ravi, Venkatasubramanian Suresh, M. Narasimha Murthy, and C. E. Veni Madhavan. Stopwords and stylometry: A latent Dirichlet allocation approach. In *Proceedings of the NIPS 2009 Workshop on Applications for Topic Models: Text and Beyond (Poster Session)*, Whistler, BC, Canada, 2009.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP 2009: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, Singapore, 2009.

Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *CSCW 1994: Proceedings of the 4th Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, NC, USA, 1994.

Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer US, 2011.

Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5(Jan):101–141, 2004.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI 2004: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494, Banff, AB, Canada, 2004.

Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information Systems*, 28(1):1–38, 2010.

Sara Rosenthal and Kathleen McKeown. Age prediction in blogs: A study of style, content, and online behavior in pre- and post-social media generations. In *ACL-HLT 2011: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 763–772, Portland, OR, USA, 2011.

Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *CICLing 2002: Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics*, pages 189–206, Mexico City, Mexico, 2002.

Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

Conrad Sanderson and Simon Guenter. Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation. In *EMNLP 2006: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 482–491, Sydney, NSW, Australia, 2006.

Ruchita Sarawgi, Kailash Gajulapalli, and Yejin Choi. Gender attribution: Tracing stylometric evidence beyond topic and genre. In *CoNLL 2011: Proceedings of the 15th International Conference on Computational Natural Language Learning*, pages 78–86, Portland, OR, USA, 2011.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW 2001: Proceedings of the 10th International World Wide Web Conference*, pages 285–295, Hong Kong, 2001.

Christina Sauper, Aria Haghighi, and Regina Barzilay. Content models with attitude. In *ACL-HLT 2011: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 350–358, Portland, OR, USA, 2011.

J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, pages 291–324. Springer-Verlag, 2007.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W. Pennebaker. Effects of age and gender on blogging. In *Proceedings of AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, pages 199–205, Stanford, CA, USA, 2006.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. Collaborative inference of sentiments from texts. In *UMAP 2010: Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization*, pages 195–206, Waikoloa, HI, USA, 2010.

Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. Personalised rating prediction for new users using latent factor models. In *HT 2011: Proceedings of the 22nd International ACM Conference on Hypertext and Hypermedia*, pages 47–56, Eindhoven, The Netherlands, 2011a.

Yanir Seroussi, Russell Smyth, and Ingrid Zukerman. Ghosts from the High Court's past: Evidence from computational linguistics for Dixon ghosting for McTiernan and Rich. *University of New South Wales Law Journal*, 34(3):984–1005, 2011b.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. Authorship attribution with latent Dirichlet allocation. In *CoNLL 2011: Proceedings of the 15th International Conference on Computational Natural Language Learning*, pages 181–189, Portland, OR, USA, 2011c.

Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. Authorship attribution with author-aware topic models. In *ACL 2012: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 264–269, Jeju Island, Republic of Korea, 2012.

Hanhuai Shan and Arindam Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *ICDM 2010: Proceedings of the 10th IEEE International Conference on Data Mining*, pages 1025–1030, Sydney, NSW, Australia, 2010.

Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the Good Grief algorithm. In *NAACL-HLT 2007: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 300–307, Rochester, NY, USA, 2007.

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. In Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, and Walter Kintsch, editors, *Handbook of Latent Semantic Analysis*, pages 427–448. Lawrence Erlbaum Associates, 2007.

Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. User-level sentiment analysis incorporating social networks. In *KDD 2011: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1397–1405, San Diego, CA, USA, 2011.

Ludovic Tanguy, Assaf Urieli, Basilio Calderone, Nabil Hathout, and Franck Sajous. A multitude of linguistically-rich features for authorship attribution. In *CLEF 2011: Proceedings of the 2011 Conference on Multilingual and Multimodal Information Access Evaluation (Lab and Workshop Notebook Papers)*, Amsterdam, The Netherlands, 2011.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476): 1566–1581, 2006.

Nava Tintarev and Judith Masthoff. Designing and evaluating explanations for recommender systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 479–510. Springer US, 2011.

Peter Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *ACL 2002: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, PA, USA, 2002.

Lyle H. Ungar and Dean P. Foster. Clustering methods for collaborative filtering. In *Proceedings of the AAAI 1998 Workshop on Recommender Systems*, pages 112–127, Madison, WI, USA, 1998.

Manolis G. Vozalis and Konstantinos G. Margaritis. Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences*, 177(15):3017–3037, 2007.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *ICML 2006: Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984, Pittsburgh, PA, USA, 2006.

Hanna M. Wallach, David Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In *NIPS 2009: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 1973–1981, Vancouver, BC, Canada, 2009a.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *ICML 2009: Proceedings of the 26th International Conference on Machine Learning*, pages 1105–1112, Montreal, QC, Canada, 2009b.

Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR 2006: Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 501–508, Seattle, WA, USA, 2006.

Xuerui Wang, Andrew McCallum, and Xing Wei. Topical N-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM 2007: Proceedings of the 7th IEEE International Conference on Data Mining*, pages 697–702, Omaha, NE, USA, 2007.

Geoffrey I. Webb, Janice R. Boughton, and Zhihai Wang. Not so naive Bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.

Sze-Meng Jojo Wong, Mark Dras, and Mark Johnson. Topic modeling for native language identification. In *ALTA 2011: Proceedings of the Australasian Language Technology Association Workshop*, pages 115–124, Canberra, ACT, Australia, 2011.

Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP 2003: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136, Sapporo, Japan, 2003.

Weishi Zhang, Guiguang Ding, Li Chen, and Chunping Li. Augmenting online video recommendations by fusing review sentiment classification. In *RSWEB 2010: Proceedings of the 2nd ACM RecSys Workshop on Recommender Systems and the Social Web*, pages 9–16, Barcelona, Spain, 2010.

Jun Zhu and Eric P. Xing. Conditional topic random fields. In *ICML 2010: Proceedings of the 27th International Conference on Machine Learning*, pages 1239–1246, Haifa, Israel, 2010.

Jun Zhu, Amr Ahmed, and Eric P. Xing. MedLDA: Maximum margin supervised topic models for regression and classification. In *ICML 2009: Proceedings of the 26th International Conference on Machine Learning*, pages 1257–1264, Montreal, QC, Canada, 2009.

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *WWW 2005: Proceedings of the 14th International World Wide Web Conference*, pages 22–32, Chiba, Japan, 2005.

Ingrid Zukerman and Diane Litman. Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, 11(1): 129–158, 2001.

Appendix A

Stopword List

This list of 571 English stopwords was obtained from `www.lextek.com/manuals/onix/stopwords2.html`.

a	amongst	away	c	could
a's	an	awfully	c'mon	couldn't
able	and	b	c's	course
about	another	be	came	currently
above	any	became	can	d
according	anybody	because	can't	definitely
accordingly	anyhow	become	cannot	described
across	anyone	becomes	cant	despite
actually	anything	becoming	cause	did
after	anyway	been	causes	didn't
afterwards	anyways	before	certain	different
again	anywhere	beforehand	certainly	do
against	apart	behind	changes	does
ain't	appear	being	clearly	doesn't
all	appreciate	believe	co	doing
allow	appropriate	below	com	don't
allows	are	beside	come	done
almost	aren't	besides	comes	down
alone	around	best	concerning	downwards
along	as	better	consequently	during
already	aside	between	consider	e
also	ask	beyond	considering	each
although	asking	both	contain	edu
always	associated	brief	containing	eg
am	at	but	contains	eight
among	available	by	corresponding	either

else	g	hers	it'd	many
elsewhere	get	herself	it'll	may
enough	gets	hi	it's	maybe
entirely	getting	him	its	me
especially	given	himself	itself	mean
et	gives	his	j	meanwhile
etc	go	hither	just	merely
even	goes	hopefully	k	might
ever	going	how	keep	more
every	gone	howbeit	keeps	moreover
everybody	got	however	kept	most
everyone	gotten	i	know	mostly
everything	greetings	i'd	knows	much
everywhere	h	i'll	known	must
ex	had	i'm	l	my
exactly	hadn't	i've	last	myself
example	happens	ie	lately	n
except	hardly	if	later	name
f	has	ignored	latter	namely
far	hasn't	immediate	latterly	nd
few	have	in	least	near
fifth	haven't	inasmuch	less	nearly
first	having	inc	lest	necessary
five	he	indeed	let	need
followed	he's	indicate	let's	needs
following	hello	indicated	like	neither
follows	help	indicates	liked	never
for	hence	inner	likely	nevertheless
former	her	insofar	little	new
formerly	here	instead	look	next
forth	here's	into	looking	nine
four	hereafter	inward	looks	no
from	hereby	is	ltd	nobody
further	herein	isn't	m	non
furthermore	hereupon	it	mainly	none

noone
nor
normally
not
nothing
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respectively
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
shouldn't
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specified
specify
specifying
still
sub
such
sup
sure
t
t's
take
taken
tell
tends
th
than
thank
thanks
thanx
that
that's
thats
the
their
theirs
them
themselves
then
thence
there
there's
thereafter
thereby
therefore
therein
theres
thereupon
these
they
they'd
they'll
they're
they've
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
u

un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
uucp
v
value
various
very
via
viz
vs
w
want
wants
was
wasn't
way
we
we'd
we'll
we're
we've
welcome
well
went
were
weren't
what
what's
whatever
when
whence
whenever
where
where's
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
who's
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
won't
wonder
would
would
wouldn't
x
y
yes
yet
you
you'd
you'll
you're
you've
your
yours
yourself
yourselves
z
zero

Appendix B

DADT Model Derivation Details

This appendix provides details on the derivation of Equation 4.5 for our DADT model (Section 4.3). As discussed in Section 4.3.2, we follow a collapsed Gibbs sampling approach to model inference, as done by Griffiths and Steyvers (2004) and Rosen-Zvi et al. (2004) for LDA and AT respectively – the two models that form the building blocks of our DADT model. This involves repeatedly sampling from the conditional distribution of the assignments to the latent variables that pertain to the current word (the di-th word, i.e., the i-th word in the d-th document), given all the other assignments (Equation 4.5):

$$p\left(x_{di}=a, y_{di}=y, z_{di}=t|\boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right) \tag{B.1}$$

where the priors and observed document authors are grouped into the set

$$\mathcal{P}=\left\{\boldsymbol{A}, \boldsymbol{\alpha}^{(D)}, \boldsymbol{\beta}^{(D)}, \delta^{(D)}, \boldsymbol{\alpha}^{(A)}, \boldsymbol{\beta}^{(A)}, \delta^{(A)}\right\} \tag{B.2}$$

It is worth noting that like Rosen-Zvi et al. (2004, 2010), we employ blocked sampling (i.e., where x_{di}, y_{di} and z_{di} are sampled together as a block), as this reduces the runtime of the sampling procedure.

Applying the definitions of conditional and joint distributions:

$$\begin{aligned}
&p\left(x_{di}=a, y_{di}=y, z_{di}=t|\boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right) = \\
&\frac{p\left(x_{di}=a, y_{di}=y, z_{di}=t, \boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}|\mathcal{P}\right)}{p\left(\boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}|\mathcal{P}\right)} = \\
&\frac{p\left(\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\mathcal{P}\right)}{p\left(w_{di}=v, \boldsymbol{W}_{-di}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}|\mathcal{P}\right)} = \\
&\frac{p\left(\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\mathcal{P}\right)}{p\left(\boldsymbol{W}_{-di}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}|\mathcal{P}\right) p\left(w_{di}=v|\boldsymbol{W}_{-di}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right)}
\end{aligned} \tag{B.3}$$

Since $p\left(w_{di} = v|\boldsymbol{W}_{-di}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right)$ is independent of the assignments of the current word:

$$p\left(x_{di} = a, y_{di} = y, z_{di} = t|\boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right) \propto \frac{p\left(\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\mathcal{P}\right)}{p\left(\boldsymbol{W}_{-di}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}|\mathcal{P}\right)} \tag{B.4}$$

Applying the joint distribution definition to the numerator:

$$p\left(\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\mathcal{P}\right) = p\left(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \mathcal{P}\right) p\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{P}\right) p\left(\boldsymbol{X}|\boldsymbol{Y}, \mathcal{P}\right) p\left(\boldsymbol{Y}|\mathcal{P}\right) \tag{B.5}$$

Doing the same for the denominator, and substituting back into Equation B.4:

$$p\left(x_{di} = a, y_{di} = y, z_{di} = t|\boldsymbol{W}, \boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right) \propto \frac{p\left(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \mathcal{P}\right)}{p\left(\boldsymbol{W}_{-di}|\boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right)} \frac{p\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{P}\right)}{p\left(\boldsymbol{Z}_{-di}|\boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \mathcal{P}\right)} \frac{p\left(\boldsymbol{X}|\boldsymbol{Y}, \mathcal{P}\right)}{p\left(\boldsymbol{X}_{-di}|\boldsymbol{Y}_{-di}, \mathcal{P}\right)} \frac{p\left(\boldsymbol{Y}|\mathcal{P}\right)}{p\left(\boldsymbol{Y}_{-di}|\mathcal{P}\right)} \tag{B.6}$$

Each of the above terms can now be solved separately by using the properties of conjugate priors, as done by Griffiths and Steyvers (2004) and Rosen-Zvi et al. (2004, 2010). We give a full example of the solution of the $\frac{p(\boldsymbol{Y}|\mathcal{P})}{p(\boldsymbol{Y}_{-di}|\mathcal{P})}$ term below. Applying similar steps to the other terms yields the following solutions (substituting these solutions and Equation B.19 into Equation B.6 yields Equation 4.5):

$$\frac{p\left(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \mathcal{P}\right)}{p\left(\boldsymbol{W}_{-di}|\boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \boldsymbol{Z}_{-di}, \mathcal{P}\right)} = \begin{cases} \dfrac{\beta_{w_{di}}^{(D)} + c_{tw_{di}}^{(DTV)}}{\sum_{v=1}^{V}\left(\beta_{v}^{(D)} + c_{tv}^{(DTV)}\right)} & \text{if } y = 0 \\ \dfrac{\beta_{w_{di}}^{(A)} + c_{tw_{di}}^{(ATV)}}{\sum_{v=1}^{V}\left(\beta_{v}^{(A)} + c_{tv}^{(ATV)}\right)} & \text{if } y = 1 \end{cases} \tag{B.7}$$

$$\frac{p\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{P}\right)}{p\left(\boldsymbol{Z}_{-di}|\boldsymbol{X}_{-di}, \boldsymbol{Y}_{-di}, \mathcal{P}\right)} = \begin{cases} \dfrac{\alpha_{t}^{(D)} + c_{dt}^{(DT)}}{\sum_{t'=1}^{T^{(D)}}\left(\alpha_{t'}^{(D)} + c_{dt'}^{(DT)}\right)} & \text{if } y = 0 \\ \dfrac{\alpha_{t}^{(A)} + c_{at}^{(AT)}}{\sum_{t'=1}^{T^{(A)}}\left(\alpha_{t'}^{(A)} + c_{at'}^{(AT)}\right)} & \text{if } y = 1 \end{cases} \tag{B.8}$$

$$\frac{p\left(\boldsymbol{X}|\boldsymbol{Y}, \mathcal{P}\right)}{p\left(\boldsymbol{X}_{-di}|\boldsymbol{Y}_{-di}, \mathcal{P}\right)} = C \tag{B.9}$$

where the count variables are defined as in Equations 4.3, 4.4 and 4.5 (i.e., excluding the di-th assignment), and C is a constant whose value is independent of the values of the assignments, since each author is drawn uniformly from the set of document authors.

Solving $\frac{p(\boldsymbol{Y}|\mathcal{P})}{p(\boldsymbol{Y}_{-di}|\mathcal{P})}$. According to the marginal and joint distribution definitions, and due to the conditional independence between documents:

$$p\left(\boldsymbol{Y}|\mathcal{P}\right) = \int_{\boldsymbol{\pi}} p(\boldsymbol{Y}, \boldsymbol{\pi}|\mathcal{P})\,\mathrm{d}\boldsymbol{\pi} = \prod_{d'=1}^{D} \int_{\pi_{d'}} p(\boldsymbol{y}_{d'}, \pi_{d'}|\mathcal{P})\,\mathrm{d}\pi_{d'} = \tag{B.10}$$

$$\prod_{d'=1}^{D} \int_{\pi_{d'}} p(\pi_{d'}|\mathcal{P}) p(\boldsymbol{y}_{d'}|\pi_{d'}, \mathcal{P})\,\mathrm{d}\pi_{d'}$$

According to DADT's definition, $\pi_{d'} \sim \text{Beta}\left(\delta^{(A)}, \delta^{(D)}\right)$. Therefore, according to the definition of the beta distribution:

$$p(\pi_{d'}|\mathcal{P}) = \frac{\pi_{d'}^{\delta^{(A)}-1}(1-\pi_{d'})^{\delta^{(D)}-1}}{B\left(\delta^{(A)}, \delta^{(D)}\right)} \tag{B.11}$$

where

$$B\left(\delta^{(A)}, \delta^{(D)}\right) = \frac{\Gamma\left(\delta^{(A)}\right)\Gamma\left(\delta^{(D)}\right)}{\Gamma\left(\delta^{(A)} + \delta^{(D)}\right)} \tag{B.12}$$

where Γ is the gamma function.

Due to the conditional independence between document words, we can rewrite $p(\boldsymbol{y}_{d'}|\pi_{d'}, \mathcal{P})$ as a product of $p(y_{d'i'}|\pi_{d'}, \mathcal{P})$ terms, and use the definition of the Bernoulli distribution to simplify this expression (because $y_{d'i'} \sim \text{Bernoulli}(\pi_{d'})$):

$$p(\boldsymbol{y}_{d'}|\pi_{d'}, \mathcal{P}) = \prod_{i'=1}^{N_{d'}} p(y_{d'i'}|\pi_{d'}, \mathcal{P}) = \prod_{i'=1}^{N_{d'}} (1-\pi_{d'})^{1-y_{d'i'}} \pi_{d'}^{y_{d'i'}} = (1-\pi_{d'})^{c_{d'}^{(DD)}} \pi_{d'}^{c_{d'}^{(DA)}} \tag{B.13}$$

where $c_{d'}^{(DD)}$ and $c_{d'}^{(DA)}$ are the counts of words assigned to document or author topics in document d', respectively (Section 4.3.2).

Substituting Equations B.11 and B.13 back into Equation B.10:

$$p\left(\boldsymbol{Y}|\mathcal{P}\right) = \tag{B.14}$$

$$\prod_{d'=1}^{D} \int_{\pi_{d'}} \frac{\pi_{d'}^{\delta^{(A)}+c_{d'}^{(DA)}-1}(1-\pi_{d'})^{\delta^{(D)}+c_{d'}^{(DD)}-1}}{B\left(\delta^{(A)}, \delta^{(D)}\right)}\,\mathrm{d}\pi_{d'} =$$

$$\prod_{d'=1}^{D} \frac{B\left(\delta^{(A)} + c_{d'}^{(DA)}, \delta^{(D)} + c_{d'}^{(DD)}\right)}{B\left(\delta^{(A)}, \delta^{(D)}\right)} \int_{\pi_{d'}} \frac{\pi_{d'}^{\delta^{(A)}+c_{d'}^{(DA)}-1}(1-\pi_{d'})^{\delta^{(D)}+c_{d'}^{(DD)}-1}}{B\left(\delta^{(A)} + c_{d'}^{(DA)}, \delta^{(D)} + c_{d'}^{(DD)}\right)}\,\mathrm{d}\pi_{d'} =$$

$$\prod_{d'=1}^{D} \frac{B\left(\delta^{(A)} + c_{d'}^{(DA)}, \delta^{(D)} + c_{d'}^{(DD)}\right)}{B\left(\delta^{(A)}, \delta^{(D)}\right)}$$

where the last step is due to the fact that the term in the integral is the probability density function of Beta $\left(\delta^{(A)} + c_{d'}^{(DA)}, \delta^{(D)} + c_{d'}^{(DD)}\right)$, which integrates to 1.

Similarly:

$$p\left(\boldsymbol{Y}_{-di}|\mathcal{P}\right) = \prod_{d'=1}^{D} \frac{B\left(\delta^{(A)} + c_{d',-di}^{(DA)}, \delta^{(D)} + c_{d',-di}^{(DD)}\right)}{B\left(\delta^{(A)}, \delta^{(D)}\right)} \tag{B.15}$$

where $c_{d',-di}^{(DD)}$ and $c_{d',-di}^{(DA)}$ are the counts of words assigned to document or author topics in document d', respectively, excluding the assignment to the di-th word.

Hence, $c_{d',-di}^{(DD)} = c_{d'}^{(DD)}$ and $c_{d',-di}^{(DA)} = c_{d'}^{(DA)}$ for all $d' \neq d$, and:

$$c_{d,-di}^{(DD)} = \begin{cases} c_d^{(DD)} - 1 & \text{if } y = 0 \\ c_d^{(DD)} & \text{if } y = 1 \end{cases} \tag{B.16}$$

$$c_{d,-di}^{(DA)} = \begin{cases} c_d^{(DA)} & \text{if } y = 0 \\ c_d^{(DA)} - 1 & \text{if } y = 1 \end{cases} \tag{B.17}$$

Therefore:

$$\frac{p\left(\boldsymbol{Y}|\mathcal{P}\right)}{p\left(\boldsymbol{Y}_{-di}|\mathcal{P}\right)} = \frac{\prod_{d'=1}^{D} \frac{B\left(\delta^{(A)}+c_{d'}^{(DA)}, \delta^{(D)}+c_{d'}^{(DD)}\right)}{B\left(\delta^{(A)},\delta^{(D)}\right)}}{\prod_{d'=1}^{D} \frac{B\left(\delta^{(A)}+c_{d',-di}^{(DA)}, \delta^{(D)}+c_{d',-di}^{(DD)}\right)}{B\left(\delta^{(A)},\delta^{(D)}\right)}} = \tag{B.18}$$

$$\prod_{d'=1}^{D} \frac{B\left(\delta^{(A)} + c_{d'}^{(DA)}, \delta^{(D)} + c_{d'}^{(DD)}\right)}{B\left(\delta^{(A)} + c_{d',-di}^{(DA)}, \delta^{(D)} + c_{d',-di}^{(DD)}\right)} = \prod_{d'=1}^{D} \frac{\frac{\Gamma\left(\delta^{(A)}+c_{d'}^{(DA)}\right)\Gamma\left(\delta^{(D)}+c_{d'}^{(DD)}\right)}{\Gamma\left(\delta^{(A)}+\delta^{(D)}+c_{d'}^{(DA)}+c_{d'}^{(DD)}\right)}}{\frac{\Gamma\left(\delta^{(A)}+c_{d',-di}^{(DA)}\right)\Gamma\left(\delta^{(D)}+c_{d',-di}^{(DD)}\right)}{\Gamma\left(\delta^{(A)}+\delta^{(D)}+c_{d',-di}^{(DA)}+c_{d',-di}^{(DD)}\right)}} =$$

$$\frac{\Gamma\left(\delta^{(A)} + c_d^{(DA)}\right)}{\Gamma\left(\delta^{(A)} + c_{d,-di}^{(DA)}\right)} \frac{\Gamma\left(\delta^{(D)} + c_d^{(DD)}\right)}{\Gamma\left(\delta^{(D)} + c_{d,-di}^{(DD)}\right)} \frac{\Gamma\left(\delta^{(A)} + \delta^{(D)} + c_{d,-di}^{(DA)} + c_{d,-di}^{(DD)}\right)}{\Gamma\left(\delta^{(A)} + \delta^{(D)} + c_d^{(DA)} + c_d^{(DD)}\right)} =$$

$$\begin{cases} \dfrac{\delta^{(D)} + c_{d,-di}^{(DD)}}{\delta^{(A)} + \delta^{(D)} + N_d - 1} & \text{if } y = 0 \\ \dfrac{\delta^{(A)} + c_{d,-di}^{(DA)}}{\delta^{(A)} + \delta^{(D)} + N_d - 1} & \text{if } y = 1 \end{cases}$$

where the last step is an application of the definition of the properties of the gamma function ($\Gamma(x+1) = x\Gamma(x)$), and since $c_d^{(DA)} + c_d^{(DD)} = N_d$ and $c_{d,-di}^{(DA)} + c_{d,-di}^{(DD)} = N_d - 1$.

Since the sum in the denominators $(\delta^{(A)} + \delta^{(D)} + N_d - 1)$ is independent of the values of the assignments, it can be omitted:

$$\frac{p\left(\boldsymbol{Y}|\mathcal{P}\right)}{p\left(\boldsymbol{Y}_{-di}|\mathcal{P}\right)} \propto \begin{cases} \delta^{(D)} + c_{d,-di}^{(DD)} & \text{if } y = 0 \\ \delta^{(A)} + c_{d,-di}^{(DA)} & \text{if } y = 1 \end{cases} \tag{B.19}$$

As discussed above, substituting Equation B.19 into Equation B.6 yields Equation 4.5 (together with the substitution of the other terms). It is worth noting that for simplicity, in Section 4.3 we used the same notation for counts that exclude the di-th elements and for counts that include the di-th elements (denoting their meaning in the text), as there were no cases where both types of counts were used in the same equation.