A Bayesian Mixed Effects Model of Literary Character

David Bamman
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dbamman@cs.cmu.edu

Ted Underwood
Department of English
University of Illinois
Urbana, IL 61801, USA
tunder@illinois.edu

Noah A. Smith
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
nasmith@cs.cmu.edu

Abstract

We consider the problem of automatically inferring latent character types in a collection of 15,099 English novels published between 1700 and 1899. Unlike prior work in which character types are assumed responsible for probabilistically generating *all* text associated with a character, we introduce a model that employs multiple effects to account for the influence of extra-linguistic information (such as author). In an empirical evaluation, we find that this method leads to improved agreement with the preregistered judgments of a literary scholar, complementing the results of alternative models.

1 Introduction

Recent work in NLP has begun to exploit the potential of entity-centric modeling for a variety of tasks: Chambers (2013) places entities at the center of probabilistic frame induction, showing gains over a comparable event-centric model (Cheung et al., 2013); Bamman et al. (2013) explicitly learn character types (or "personas") in a dataset of Wikipedia movie plot summaries; and entity-centric models form one dominant approach in coreference resolution (Durrett et al., 2013; Haghighi and Klein, 2010).

One commonality among all of these very different probabilistic approaches is that each learns statistical regularities about how entities are depicted in text (whether for the sake of learning a set of semantic roles, character types, or linking anaphora to the entities to which they refer). In each case, the text we observe associated with an entity in a document is directly dependent on the class of entity—and only that class. This relationship between entity and text is a theoretical assumption, with important consequences for learning: entity types learned in this way will be increasingly similar the more similar the domain, author, and other extra-linguistic effects are between them.[1] While in many cases the topically similar types learned under this assumption may be desirable, we explore here the alternative, in which entity types are learned in a way that controls for such effects. In introducing a model based on different assumptions, we provide a method that complements past work and provides researchers with more flexible tools to infer different kinds of character types.

We focus here on the literary domain, exploring a large collection of 15,099 English novels published in the 18th and 19th centuries. By accounting for the influence of individual authors while inferring latent character types, we are able to learn personas that cut across different authors more effectively than if we learned types conditioned on the text alone. Modeling the language used to describe a character as the joint result of that character's latent type and of other formal variables allows us to test multiple models of character and assess their value for different interpretive problems. As a test case, we focus on separating character from authorial diction, but this approach can readily be generalized to produce models that provisionally distinguish character from other factors (such as period, genre, or point of view) as well.

2 Literary Background

Inferring character is challenging from a literary perspective partly because scholars have not reached consensus about the meaning of the term. It may seem obvious that a "character" is a representation of a (real or imagined) person, and many humanists do use the term that way. But there is

[1] For example, many entities in Early Modern English texts may be judged to be more similar to each other than to entities from later texts simply by virtue of using *hath* and other archaic verb forms.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 370–379,
Baltimore, Maryland, USA, June 23-25 2014.

an equally strong critical tradition that treats character as a formal dimension of narrative. To describe a character as a "blocking figure" or "first-person narrator," for instance, is a statement less about the attributes of an imagined person than about a narrative function (Keen, 2003). Characters are in one sense collections of psychological or moral attributes, but in another sense "word-masses" (Forster, 1927). This tension between "referential" and "formalist" models of character has been a centrally "divisive question in ... literary theory" (Woloch, 2003).

Considering primary source texts (as distinct from plot summaries) forces us to confront new theoretical questions about character. In a plot summary (such as those explored by Bamman et al., 2013), a human reader may already have used implicit models of character to extract high-level features. To infer character types from raw narrative text, researchers need to explicitly model the relationship of character to narrative form. This is not a solved problem, even for human readers.

For instance, it has frequently been remarked that the characters of Charles Dickens share certain similarities—including a reliance on tag phrases and recurring tics. A referential model of character might try to distinguish this common stylistic element from underlying "personalities." A strictly formalist model might refuse to separate authorial diction from character at all. In practice, human readers can adopt either perspective: we recognize that characters have a "Dickensian" quality but also recognize that a Dickens villain is (in one sense) more like villains in other authors than like a Dickensian philanthropist. Our goal is to show that computational methods can support the same range of perspectives—allowing a provisional, flexible separation between the referential and formal dimensions of narrative.

3 Data

The dataset for this work consists of 15,099 distinct narratives drawn from HathiTrust Digital Library.[2] From an initial collection of 469,200 volumes written in English and published between 1700 and 1899 (including poetry, drama, and nonfiction as well as prose narrative), we extract 32,209 volumes of prose fiction, remove duplicates and fuse multi-volume works to create the final dataset. Since the original texts were produced by scanning and running OCR on physical books, we automatically correct common OCR errors and trim front and back matter from the volumes using the page-level classifiers and HMM of Underwood et al. (2013)

Many aspects of this process would be simpler if we used manually-corrected texts, such as those drawn from Project Gutenberg. But we hope to produce research that has historical as well as computational significance, and doing so depends on the provenance of a collection. Gutenberg's decentralized selection process tends to produce exceptionally good coverage of currently-popular genres like science fiction, whereas HathiTrust aggregates university libraries. Library collections are not guaranteed to represent the past perfectly, but they are larger, and less strongly shaped by contemporary preferences.

The goal of this work is to provide a method to infer a set of character types in an unsupervised fashion from the data. As with prior work (Bamman et al., 2013), we define this target, a character *persona*, as a distribution over several categories of typed dependency relations:[3]

1. agent: the actions of which a character is the agent (i.e., verbs for which the character holds an nsubj or agent relation).
2. patient: the actions of which a character is the patient (i.e., verbs for which the character holds a dobj or nsubjpass relation).
3. possessive: the objects that a character possesses (i.e., all words for which the character holds a poss relation).
4. predicative: attributes predicated of a character (i.e., adjectives or nouns holding an nsubj relation to the character, with an inflection of *be* as a child).

This set captures the constellation of what a character *does* and *has done to them*, what they *possess*, and what they are described as *being*.

While previous work uses the Stanford CoreNLP toolkit to identify characters and extract typed dependencies for them, we found this approach to be too slow for the scale of our data (a total of 1.8 billion tokens); in particular, syntactic parsing, with cubic complexity in sentence length, and out-of-the-box coreference resolution (with thousands of potential antecedents) prove to be

[2] http://www.hathitrust.org

[3] All categories are described using the Stanford typed dependencies (de Marneffe and Manning, 2008), but any syntactic formalism is equally applicable.

the biggest bottlenecks.

Before addressing character inference, we present here a prerequisite NLP pipeline that scales well to book-length documents.[4] This pipeline uses the Stanford POS tagger (Toutanova et al., 2003), the linear-time MaltParser (Nivre et al., 2007) for dependency parsing (trained on Stanford typed dependencies), and the Stanford named entity recognizer (Finkel et al., 2005). It includes the following components for clustering character name mentions, resolving pronominal coreference, and reducing vocabulary dimensionality.

3.1 Character Clustering

First, let us terminologically distinguish between a character *mention* in a text (e.g., the token *Tom* on page 141 of *The Adventures of Tom Sawyer*) and a character *entity* (e.g., TOM SAWYER the character, to which that token refers). To resolve the former to the latter, we largely follow Davis et al. (2003) and Elson et al. (2010): we define a set of initial characters corresponding to each unique character name that is not a subset of another (e.g., *Mr. Tom Sawyer*) and deterministically create a set of allowable variants for each one (*Mr. Tom Sawyer* → *Tom*, *Sawyer*, *Tom Sawyer*, *Mr. Sawyer*, and *Mr. Tom*); then, from the beginning of the book to the end, we greedily assign each mention to the most recently linked entity for whom it is a variant. The result constitutes our set of characters, with all mentions partitioned among them.

3.2 Pronominal Coreference Resolution

While the character clustering stage is essentially performing proper noun coreference resolution, approximately 74% of references to characters in books come in the form of pronouns.[5] To resolve this more difficult class at the scale of an entire book, we train a log-linear discriminative classifier only on the task of resolving *pronominal* anaphora (i.e., ignoring generic noun phrases such as *the paint* or *the rascal*).

For this task, we annotated a set of 832 coreference links in 3 books (*Pride and Prejudice*, *The Turn of the Screw*, and *Heart of Darkness*) and featurized coreference/antecedent pairs with:

1. The syntactic dependency path from a pronoun to its potential antecedent (e.g., dobj↑pred→↓pred↓nsubj (where → denotes movement across sentence boundaries).
2. The salience of the antecedent character (defined as the count of that character's named mentions in the previous 500 words).
3. The antecedent part of speech.
4. Whether or not the pronoun and antecedent appear in the same quotation scope (false if one appears in a quotation and one outside).
5. Whether or not the two agree for gender.
6. The syntactic tree distance between the two.
7. The linear (word) distance between the two.

With this featurization and training data, we train a binary logistic regression classifier with ℓ_1 regularization (where negative examples are comprised of all character entities in the previous 100 words not labeled as the true antecedent). In a 10-fold cross-validation on predicting the true nearest antecedent for a pronominal anaphor, this method achieves an average accuracy of 82.7%.

With this trained model, we then select the highest-scoring antecedent within 100 words for each pronominal anaphor in our data.

3.3 Dimensionality Reduction

To manage the degrees of freedom in the model described in §4, we perform dimensionality reduction on the vocabulary by learning word embeddings with a log-linear continuous skip-gram language model (Mikolov et al., 2013) on the entire collection of 15,099 books. This method learns a low-dimensional real-valued vector representation of each word to predict all of the words in a window around it; empirically, we find that with a sufficient window size (we use $n = 10$), these word embeddings capture semantic similarity (placing topically similar words near each other in vector space).[6] We learn a 100-dimensional embedding for each of the 512,344 words in our vocabulary.

To create a partition over the vocabulary, we use hard K-means clustering (with Euclidean distance) to group the 512,344 word types into 1,000 clusters. We then agglomeratively cluster those 1,000 groups to assign bitstring representations to each one, forming a balanced binary tree by only merging existing clusters at equal levels in the hi-

[4] All code is available at `http://www.ark.cs.cmu.edu/literaryCharacter`

[5] Over all 15,099 narratives, the average number of character proper name mentions is 1,673; the average number of gendered singular pronouns (*he, she, him, his, her*) is 4,641.

[6] In comparison, Brown et al. (1992) clusters learned from the same data capture *syntactic* similarity (placing functionally similar words in the same cluster).

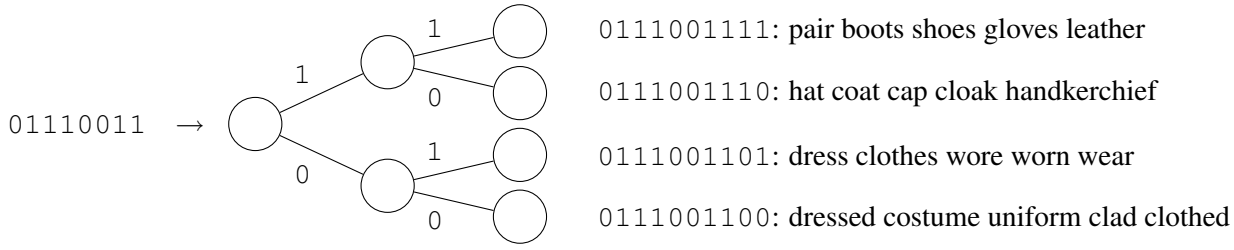


Figure 1: Bitstring representations of neural agglomerative clusters, illustrating the leaf nodes in a binary tree rooted in the prefix `01110011`. Bitstring encodings of intermediate nodes and terminal leaves result by following the left (`0`) and right (`1`) branches of the merge tree created through agglomerative clustering.

erarchy. We use Euclidean distance as a fundamental metric and a group-average similarity function for calculating the distance between groups. Fig. 1 illustrates four of the 1,000 learned clusters.

4 Model

In order to separate out the effects that a character's persona has on the words that are associated with them (as opposed to other factors, such as time period, genre, or author), we adopt a hierarchical Bayesian approach in which the words we observe are generated conditional on a combination of different effects captured in a log-linear (or "maximum entropy") distribution.

Maximum entropy approaches to language modeling have been used since Rosenfeld (1996) to incorporate long-distance information, such as previously-mentioned trigger words, into n-gram language models. This work has since been extended to a Bayesian setting by applying both a Gaussian prior (Chen and Rosenfeld, 2000), which dampens the impact of any individual feature, and sparsity-inducing priors (Kazama and Tsujii, 2003; Goodman, 2004), which can drive many feature weights to 0. The latter have been applied specifically to the problem of estimating word probabilities with sparse additive generative (SAGE) models (Eisenstein et al., 2011), where sparse extra-linguistic effects can influence a word probability in a larger generative setting.

In contrast to previous work in which the probability of a word linked to a character is dependent entirely on the character's latent persona, in our model, we see the probability of a word as dependent on: (i) the **background** likelihood of the word, (ii) the **author**, so that a word becomes more probable if a particular author tends to use it more, and (iii) the character's **persona**, so that a word is more probable if appearing with a particular persona. Intuitively, if the author *Jane Austen* is associated with a high weight for the word *manners*, and all personas have little effect for this word, then *manners* will have little impact on deciding which persona a particular Austen character embodies, since its presence is explained largely by Austen having penned the word. While we address only the author as an observed effect, this model is easily extended to other features as well, including period, genre, point of view, and others.

The generative story runs as follows (Figure 2 depicts the full graphical model): Let there be M unique authors in the data, P latent personas (a hyperparameter to be set), and V words in the vocabulary (in the general setting these may be word types; in our data the vocabulary is the set of 1,000 unique cluster IDs). Each role type $r \in \{\textsf{agent}, \textsf{patient}, \textsf{possessive}, \textsf{predicative}\}$ and vocabulary word v (here, a cluster ID) is associated with a real-valued vector $\eta_{r,v} = [\eta^{meta}_{r,v}, \eta^{pers}_{r,v}, \eta^{0}_{r,v}]$ of length $M + P + 1$. The first $M + P$ elements are drawn from a Laplace prior with mean $\mu = 0$ and scale $\lambda = 1$; the last element $\eta^{0}_{r,v}$ is an unregularized bias term accounting for the background. Each element in this vector captures the log-additive effect of each author, persona, and the background distribution on the word's probability (Eq. 1, below).

Much like latent Dirichlet allocation (Blei et al., 2003), each document d in our dataset draws a multinomial distribution θ_d over personas from a shared Dirichlet prior α, which captures the proportion of each character type in that particular document. Every character c in the document draws its persona p from this document-specific multinomial. Given document metadata m (here, one of a set of M authors) and persona p, each tuple of a role r with word w is assumed to be drawn from Eq. 1 in Fig. 3. This SAGE model can be understood as a log-linear distribution with three kinds of features (metadata, persona, and back-

$$P(w \mid m, p, r, \eta) = \exp\left(\eta^{meta}_{r,w}[m] + \eta^{pers}_{r,w}[p] + \eta^{0}_{r,w}\right) \Big/ \sum_{v=1}^{V} \exp\left(\eta^{meta}_{r,v}[m] + \eta^{pers}_{r,v}[p] + \eta^{0}_{r,v}\right) \quad (1)$$

$$P(b \mid m, p, r, \eta) = \prod_{j=0}^{n-1} \begin{cases} \text{logit}^{-1}\left(\eta^{meta}_{r,b_{1:j}}[m] + \eta^{pers}_{r,b_{1:j}}[p] + \eta^{0}_{r,b_{1:j}}\right) & \text{if } b_{j+1} = 1 \\ 1 - \text{logit}^{-1}\left(\eta^{meta}_{r,b_{1:j}}[m] + \eta^{pers}_{r,b_{1:j}}[p] + \eta^{0}_{r,b_{1:j}}\right) & \text{otherwise} \end{cases} \quad (2)$$

Figure 3: Parameterizations of the SAGE word distribution. Eq. 1 is a "flat" multinomial logistic regression with one η-vector per role and word. Eq. 2 uses the hierarchical softmax formulation, with one η-vector per role and node in the binary tree of word clusters, giving a distribution over bit strings (b) with the same number of parameters as Eq. 1.

ground bias).

4.1 Hierarchical Softmax

The partition function in Eq. 1 can lead to slow inference for any reasonably-sized vocabulary. To address this, we reparameterize the model by exploiting the structure of the agglomerative clustering in §3.3 to perform a hierarchical softmax, following Goodman (2001), Morin and Bengio (2005) and Mikolov et al. (2013).

The bitstring representations by which we encode each word in the vocabulary serve as natural, and inherently meaningful, intermediate classes that correspond to semantically related subsets of the vocabulary, with each bitstring prefix denoting one such class. Longer bitstrings correspond to more fine-grained classes. In the example shown in Figure 1, `011100111` is one such intermediate class, containing the union of *pair, boots, shoes, gloves leather* and *hat, coat, cap cloak, handkerchief*. Because these classes recursively partition the vocabulary, they offer a convenient way to reparameterize the model through the chain rule of probability.

Consider, for example, a word represented as the bitstring $c = \texttt{01011}$; calculating $P(c = \texttt{01011})$—we suppress conditioning variables for clarity—involves the product: $P(c_1 = \texttt{0}) \times P(c_2 = \texttt{1} \mid c_1 = \texttt{0}) \times P(c_3 = \texttt{0} \mid c_{1:2} = \texttt{01}) \times P(c_4 = \texttt{1} \mid c_{1:3} = \texttt{010}) \times P(c_5 = \texttt{1} \mid c_{1:4} = \texttt{0101})$.

Since each multiplicand involves a binary prediction, we can avoid partition functions and use the classic binary logistic regression.[7] We have converted the V-way multiclass logistic regression problem of Eq. 1 into a sequence of $\log V$ evaluations (assuming a perfectly balanced tree). Given m, p, and r (as above) we let $b = b_1 b_2 \cdots b_n$ denote the bitstring representation of a word cluster, and the distribution is given by Eq. 2 in Fig. 3.

In this paramaterization, rather than one η-vector for each role and vocabulary term, we have one η-vector for each role and conditional binary decision in the tree (each bitstring prefix). Since the tree is binary with V leaves, this yields the same total number of parameters. As Goodman (2001) points out, while this reparameterization is exact for true probabilities, it remains an approximation for estimated models (with generalization behavior dependent on how well the class hierarchy is supported by the data). In addition to enabling faster inference, one advantage of the bitstring representation and the hierarchical softmax parameterization is that we can easily calculate probabilities of clusters at different granularities.

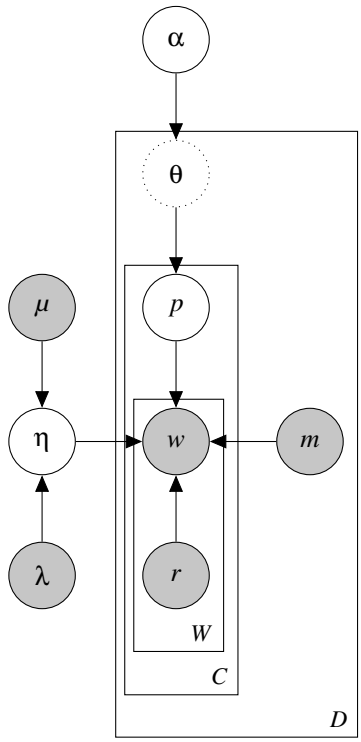
4.2 Inference

Our primary quantities of interest in this model are p (the personas for each character) and η, the effects that each author and persona have on the probability of a word. Rather than adopting a fully Bayesian approach (e.g., sampling all variables), we infer these values using stochastic EM, alternating between collapsed Gibbs sampling for each p and maximizing with respect to η.

Collapsed Gibbs for personas.[8] At each step, the required quantity is the probability that character c in document d has persona z, given everything else. This is proportional to the number of other characters in document d who also (currently) have that persona (plus the Dirichlet hyperparameter which acts as a smoother) times the probability (under $p_{d,c} = z$) of all of the words

[7] Recall that logistic regression lets $P_{LR}(y = 1 \mid x, \beta) = \text{logit}^{-1}(x^\top \beta) = 1/(1 + \exp -x^\top \beta)$ for binary dependent variable y, independent variables x, and coefficients β.

[8] We assume the reader is familiar with collapsed Gibbs sampling as used in latent-variable NLP models.



P	Number of personas (hyperparameter)
D	Number of documents
C_d	Number of characters in document d
$W_{d,c}$	Number of (cluster, role) tuples for character c
m_d	Metadata for document d (ranges over M authors)
θ_d	Document d's distribution over personas
$p_{d,c}$	Character c's persona
j	An index for a $\langle r, w \rangle$ tuple in the data
w_j	Word cluster ID for tuple j
r_j	Role for tuple $j \in \{\textsf{agent}, \textsf{patient}, \textsf{poss}, \textsf{pred}\}$
η	Coefficients for the log-linear language model
μ, λ	Laplace mean and scale (for regularizing η)
α	Dirichlet concentration parameter

Figure 2: **Above:** Probabilistic graphical model. Observed variables are shaded, latent variables are clear, and collapsed variables are dotted. **Below:** Definition of variables.

observed in each role r for that character:

$$(count(z; p_{d,-c}) + \alpha_z) \times \prod_{r=1}^{R} \prod_{j:r_j=r} P(b_j \mid m, p, r, \eta) \quad (3)$$

The metadata features (like author, etc.) influence this probability by being constant for all choices of z; e.g., if the coefficient learned for *Austen* for vocabulary term *manners* is high and all coefficients for all z are close to zero, then the probability of *manners* will change little under different choices of z. Eq. 3 contains one multiplicand for every word associated with a character, and only one term reflecting the influence of the shared document multinomial. The implication is that, for major characters with many observed words, the words will dominate the choice of persona; where the document influence would have a bigger effect is with characters for whom we don't have much data. In that case, it can act as a kind of informed background; given what little data we have for that character, it would nudge us toward the character types that the other characters in the book embody.

Given an assignment of all p, we choose η to maximize the conditional log-likelihood of the words, as represented by their bitstring cluster IDs, given the observed author and background effects and the sampled personas. This equates to solving $4V$ ℓ_1-regularized logistic regressions (see Eq. 2 in Figure 3), one for each role type and bitstring prefix, each with $M + P + 1$ parameters. We apply OWL-QN (Andrew and Gao, 2007) to minimize the ℓ_1-regularized objective with an absolute convergence threshold of 10^{-5}.

5 Evaluation

While standard NLP and machine learning practice is to evaluate the performance of an algorithm on a held-out gold standard, articulating what a true "persona" might be for a character is inherently problematic. Rather, we evaluate the performance and output of our model by preregistering a set of 29 hypotheses of varying scope and difficulty and comparing the performance of different models in either confirming, or failing to confirm, those hypotheses. This kind of evaluation was previously applied to a subjective text measurement problem by Sim et al. (2013).

All hypotheses were created by a literary scholar with specialization in the period to not only give an empirical measure of the strengths and weaknesses of different models, but also to help explore exactly what the different models may, or may not, be learning. All preregistered hypotheses establish the degrees of similarity among three characters, taking the form: "character X is more similar to character Y than either X or Y is to a distractor character Z"; for a given model and definition of distance under that model, each hypothesis yields two yes/no decisions that we can evaluate:

- $distance(X, Y) < distance(X, Z)$
- $distance(X, Y) < distance(Y, Z)$

To tease apart the different kinds of similarities we hope to explore, we divide the hypotheses into four classes:

A. This class constitutes **sanity checks**: character X and Y are more similar to each other in every way than to character Z. E.g.: Elizabeth Bennet in *Pride and Prejudice* resembles Elinor Dashwood in *Sense and Sensibility* (Jane Austen) more than either character resembles Allen Quatermain in *Allen Quatermain* (H. Rider Haggard). (Austenian protagonists should resemble each other more than they resemble a grizzled hunter.)
B. This class captures our ability to identify two characters in the same author as being more similar to each other than to a closely related character in a **different** author. E.g.: Wickham in *Pride and Prejudice* resembles Willoughby in *Sense and Sensibility* (Jane Austen) more than either character resembles Mr. Rochester in *Jane Eyre* (Charlotte Brontë).
C. This class captures our ability to discriminate among similar characters in the **same** author. In these hypotheses, two characters X and Y from the same author are more similar to each other than to a third character Z from that same author. E.g.: Wickham in *Pride and Prejudice* (Jane Austen) resembles Willoughby in *Sense and Sensibility* more than either character resembles Mr. Darcy in *Pride and Prejudice.*
D. This class constitutes more difficult, **exploratory** hypotheses, including differences among point of view. E.g.: Montoni in *Mysteries of Udolpho* (Radcliffe) resembles Heathcliff in *Wuthering Heights* (Emily Brontë) more than either resembles Mr. Bennet in *Pride and Prejudice.* (Testing our model's ability to discern similarities in spite of elapsed time.)

All 29 hypotheses can be found in a supplementary technical report (Bamman et al., 2014). We emphasize that the full set of hypotheses was locked *before* the model was estimated.

6 Experiments

Part of the motivation of our mixed effects model is to be able to tackle hypothesis class C—by factoring out the influence of a particular author on the learning of personas, we would like to be able to discriminate between characters that all have a common authorial voice. In contrast, the Persona Regression model of Bamman et al. (2013), which uses metadata variables (like authorship) to encourage entities with similar covariates to have similar personas, reflects an assumption that makes it likely to perform well at class B.

To judge their respective strengths on different hypothesis classes, we evaluate three models:

1. The mixed-effects Author/Persona model (described above), which includes author information as a metadata effect; here, each η-vector (of length $M + P + 1$) contains a parameter for each of the distinct authors in our data, a parameter for each persona, and a background parameter.
2. A Basic persona model, which ablates author information but retains the same log-linear architecture; here, the η-vector is of size $P + 1$ and does not model author effects.
3. The Persona Regression model of Bamman et al. (2013).

All models are run with $P \in \{10, 25, 50, 100, 250\}$ personas; Persona Regression additionally uses $K = 25$ latent topics. All configurations use the full dataset of 15,099 novels, and all characters with at least 25 total roles (a total of 257,298 entities). All experiments are run with 50 iterations of Gibbs sampling to collect samples for the personas p, alternating with maximization steps for η. The value of α is optimized using slice sampling (with a non-informative prior) every 5 iterations. The value of λ is held constant at 1. At the end of inference, we calculate the posterior distributions over personas for all characters as the sampling probability of the final iteration.

To formally evaluate "similarity" between two characters, we measure the Jensen-Shannon divergence between personas (calculated as the average JS distance over the cluster distributions for each role type), marginalizing over the characters' posterior distributions over personas; two characters with a lower JS divergence are judged to be more similar than two characters with a higher one.

As a Baseline, we also evaluate all hypotheses on a model with no latent variables whatsoever, which instead measures similarity as the average JS divergence between the empirical word distributions over each role type.

Table 1 presents the results of this comparison; for all models with latent variables, we report the average of 5 sampling runs with different random initializations. Figure 4 provides a syn-

P	Model	Hypothesis Class A	B	C	D
250	Author/Persona	1.00	0.58	0.75	0.42
	Basic Persona	1.00	0.73	0.58	0.53
	Persona Reg.	0.90	0.70	0.58	0.44
100	Author/Persona	0.98	0.68	0.70	0.46
	Basic Persona	0.95	0.73	0.53	0.47
	Persona Reg.	0.93	0.78	0.63	0.49
50	Author/Persona	0.95	0.73	0.63	0.50
	Basic Persona	0.98	0.75	0.48	0.53
	Persona Reg.	1.00	0.75	0.65	0.38
25	Author/Persona	1.00	0.63	0.65	0.50
	Basic Persona	1.00	0.63	0.50	0.50
	Persona Reg.	0.90	0.78	0.60	0.39
10	Author/Persona	0.95	0.63	0.70	0.51
	Basic Persona	0.78	0.80	0.48	0.46
	Persona Reg.	0.90	0.73	0.43	0.41
Baseline		1.00	0.63	0.58	0.37

Table 1: Agreement rates with preregistered hypotheses, averaged over 5 sampling runs with different initializations.

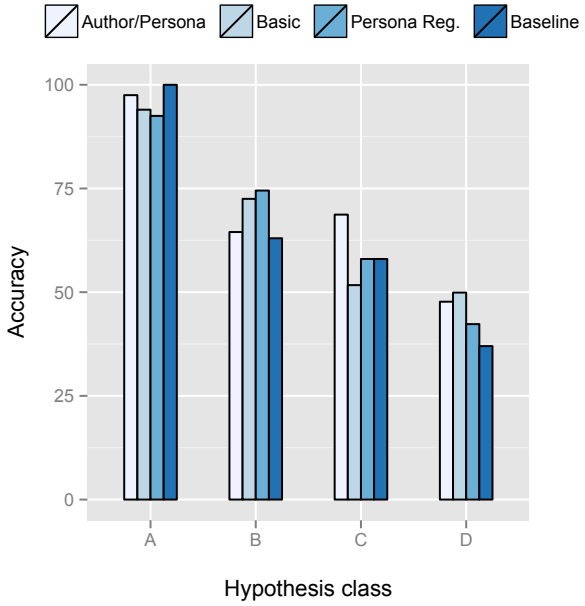


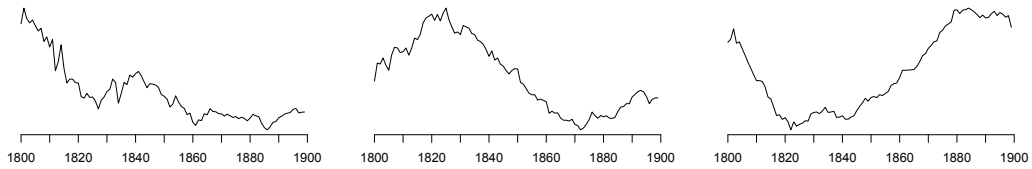
Figure 4: Synopsis of table 1: average accuracy across all P. Persona regression is best able to judge characters in one author to be more similar to each other than to characters in another (B), while our mixed-effects Author/Persona model outperforms other models at discriminating characters in the same author (C).

opsis of this table by illustrating the average accuracy across all choice of P. All models, including the baseline, perform well on the sanity checks (A). As expected, the Persona Regression model performs best at hypothesis class B (correctly judging two characters from the same author to be more similar to each other than to a character from a different author); this behavior is encouraged in this model by allowing an author (as an external metadata variable) to directly influence the persona choice, which has the effect of pushing characters from the same author to embody the same character type. Our mixed effects Author/Persona model, in contrast, outperforms the other models at hypothesis class C (correctly discriminating different character types present in the same author). By discounting author-specific lexical effects during persona inference, we are better able to detect variation among the characters of a single author that we are not able to capture otherwise. While these different models complement each other in this manner, we note that there is no absolute separation among them, which may be suggestive of the degree to which the formal and referential dimensions are fused in novels. Nevertheless, the strengths of these different models on these different hypothesis classes gives us flexible alternatives to use depending on the kinds of character types we are looking to infer.

7 Analysis

The latent personas inferred from this model will support further exploratory analysis of literary history. Figure 2 illustrates this with a selection of three character types learned, displaying characteristic clusters for all role types, along with the distribution of that persona's use across time and the gender distribution of characters embodying that persona. In general, the personas learned so far do not align neatly with character types known to literary historians. But they do have legible associations both with literary genres and with social categories. Even though gender is not an observable variable known to the model during inference, personas tend to be clearly gendered. This is not in itself surprising (since literary scholars know that assumptions about character are strongly gendered), but it does suggest that diachronic analysis of latent character types might cast new light on the history of gender in fiction. This is especially true since the distribution of personas across the time axis similarly reveals coherent trends.

Table 3 likewise illustrates what our model learns by presenting a sample of the fixed effects learned for a set of five major 19th-century authors. These are clusters that are conditionally more likely to appear associated with a character in a work by the given author than they are in the overall data; by factoring this information out of the inference process for learning character types (by attributing its presence in a text to the author



Agent	carried ran threw rose fell suddenly is seems	sent received arrived appeared struck showed returned immediately waiting	turns begins returns thinks loves calls does knows comes
Patient	wounded killed murdered suffer yield acknowledge free saved unknown	wounded killed murdered destroy bind crush attend haste proceed	thinks loves calls love hope true turn hold show
Poss	death happiness future lips cheek brow mouth fingers tongue	army officers troops soldiers band armed party join camp	lips cheek brow eyes face eye table bed chair
Pred	crime guilty murder youth lover hers dead living died	king emperor throne general officer guard soldier knight hero	beautiful fair fine good kind ill dead living died
% Female	12.2	3.7	54.7

Table 2: Snapshots of three personas learned from the $P = 50$, Author/Persona model. Gender and time proportions are calculated by summing and normalizing the posterior distributions over all characters with that feature. We truncate time series at 1800 due to data sparsity before that date; the y-axis illustrates the frequency of its use in a given year, relative to its lifetime.

Author	clusters
Jane Austen	praise gift consolation letter read write character natural taste
Charlotte Brontë	lips cheek brow book paper books hat coat cap
Charles Dickens	hat coat cap table bed chair hand head hands
Herman Melville	boat ship board hat coat cap feet ground foot
Jules Verne	journey travel voyage master company presence success plan progress

Table 3: Characteristic possessive clusters in a sample of major 19th-century authors.

rather than the persona), we are able to learn personas that cut across different topics more effectively than if a character type is responsible for explaining the presence of these terms as well.

8 Conclusion

Our method establishes the possibility of representing the relationship between character and narrative form in a hierarchical Bayesian model. Postulating an interaction between authorial diction and character allows models that consider the effect of the author to more closely reproduce a human reader's judgments, especially by learning to distinguish different character types within a single author's oeuvre. This opens the door to considering other structural and formal dimensions of narration. For instance, representation of character is notoriously complicated by narrative point of view (Booth, 1961); and indeed, comparisons between first-person narrators and other characters are a primary source of error for all models tested above. The strategy we have demonstrated suggests that it might be productive to address this by modeling the interaction of character and point of view as a separate effect analogous to authorship.

It is also worth noting that the models tested above diverge from many structuralist theories of narrative (Propp, 1998) by allowing multiple instances of the same persona in a single work. Learning structural limitations on the number of "protagonists" likely to coexist in a single story, for example, may be another fruitful area to explore. In all cases, the machinery of hierarchical models gives us the flexibility to incorporate such effects at will, while also being explicit about the theoretical assumptions that attend them.

9 Acknowledgments

We thank the reviewers for their helpful comments. The research reported here was supported by a National Endowment for the Humanities start-up grant to T.U., U.S. National Science Foundation grant CAREER IIS-1054319 to N.A.S., and an ARCS scholarship to D.B. This work was made possible through the use of computing resources made available by the Pittsburgh Supercomputing Center. Eleanor Courtemanche provided advice about the history of narrative theory.

References

Galen Andrew and Jianfeng Gao. 2007. Scalable training of l_1-regularized log-linear models. In *Proc. of ICML*.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. *Proc. of ACL*.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. Appendix to 'A Bayesian mixed effects model of literary character'. Technical report, Carnegie Mellon University, University of Illinois-Urbana Champaign.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Wayne Booth. 1961. *The Rhetoric of Fiction*. University of Chicago Press, Chicago.

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

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proc. of EMNLP*, Seattle, Washington, USA.

Stanley F. Chen and Roni Rosenfeld. 2000. A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proc. of NAACL*.

Peter T. Davis, David K. Elson, and Judith L. Klavans. 2003. Methods for precise named entity matching in digital collections. In *Proc. of JCDL*, Washington, DC, USA.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. Technical report, Stanford University.

Greg Durrett, David Hall, and Dan Klein. 2013. Decentralized entity-level modeling for coreference resolution. In *Proc. of ACL*.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *Proc. of ICML*.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. In *Proc. of ACL*, Stroudsburg, PA, USA.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL*.

E. M. Forster. 1927. *Aspects of the Novel*. Harcourt, Brace & Co.

Joshua Goodman. 2001. Classes for fast maximum entropy training. In *Proc. of ICASSP*.

Joshua Goodman. 2004. Exponential priors for maximum entropy models. In *Proc. of NAACL*.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Proc. of NAACL*.

Jun'ichi Kazama and Jun'ichi Tsujii. 2003. Evaluation and extension of maximum entropy models with inequality constraints. In *Proc. of EMNLP*.

Suzanne Keen. 2003. *Narrative Form.* Palgrave Macmillan, Basingstoke.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proc. of ICLR*.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proc. of AISTATS*.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135, 5.

Vladimir Propp. 1998. *Morphology of the Folktale*. University of Texas Press, 2nd edition.

Roni Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language*, 10(3):187 – 228.

Yanchuan Sim, Brice D. L. Acree, Justin H. Gross, and Noah A. Smith. 2013. Measuring ideological proportions in political speeches. In *Proc. of EMNLP*, Seattle, Washington, USA.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of NAACL*.

Ted Underwood, Michael L Black, Loretta Auvil, and Boris Capitanu. 2013. Mapping mutable genres in structurally complex volumes. In *Proc. of IEEE International Conference on Big Data*.

Alex Woloch. 2003. *The One vs. the Many: Minor Characters and the Space of the Protagonist in the Novel*. Princeton University Press, Princeton NJ.