Improving Quotation Attribution with Fictional Character Embeddings

Gaspard Michel[†*]
gmichel@deezer.com

Elena V. Epure[†]
eepure@deezer.com

Romain Hennequin[†]
rhennequin@deezer.com

Christophe Cerisara[*]
christophe.cerisara@loria.fr

[†] Deezer Research, Paris, France
[*] LORIA, CNRS, Nancy, France

Abstract

Humans naturally attribute utterances of direct speech to their speaker in literary works. When attributing quotes, we process contextual information but also access mental representations of characters that we build and revise throughout the narrative. Recent methods to automatically attribute such utterances have explored simulating human logic with deterministic rules or learning new implicit rules with neural networks when processing contextual information. However, these systems inherently lack *character* representations, which often leads to errors in more challenging examples of attribution: anaphoric and implicit quotes. In this work, we propose to augment a popular quotation attribution system, BookNLP, with character embeddings that encode global stylistic information of characters derived from an off-the-shelf stylometric model, Universal Authorship Representation (UAR). We create DramaCV[1], a corpus of English drama plays from the 15th to 20th century that we automatically annotate for Authorship Verification of fictional characters utterances, and release two versions of UAR trained on DramaCV, that are tailored for literary characters analysis. Then, through an extensive evaluation on 28 novels, we show that combining BookNLP's contextual information with our proposed global character embeddings improves the identification of speakers for anaphoric and implicit quotes, reaching state-of-the-art performance.

1 Introduction

Authors of literary works employ various devices to create engaging narratives, often combining narration with character dialogues to unveil the plot. Fictional characters portray themselves through dialogues, revealing aspects of their personality, their own style and ideas about themselves and the fictional world. As part of studying characters in digi-

"As soon as ever Mr. Bingley comes, my dear," said Mrs. Bennet, "you will wait on him of course."

"No, no. You forced me into visiting him last year, and promised if I went to see him, he should marry one of my daughters..."

His wife represented to him how absolutely necessary such an attention would be from all the neighbouring gentlemen, on his returning to Netherfield.

"'Tis an etiquette I despise," said he.

Figure 1: Excerpt of *Pride and Prejudice* by Jane Austen (1813). Quotations are colored by quote type: explicit, implicit and anaphoric. Speaker information given by the narrator are underlined.

tal humanities, automatically identifying utterances and attributing them to characters, also known as quotation attribution, is central (Elson et al., 2010; Muzny et al., 2017a; Labatut and Bost, 2019; Sims and Bamman, 2020).

Recent approaches to quotation attribution typically succeed at attributing *explicit* quotes but struggle when it comes to *anaphoric* and *implicit* quotes (Muzny et al., 2017b; Cuesta-Lazaro et al., 2022; Vishnubhotla et al., 2023; Su et al., 2023). Explicit utterances occur when the narrator indicates the speaker of a quote with a speech verb and a named mention, while anaphoric quotes are introduced with a speech verb and a pronoun or common noun. When no narrative information is given about the speaker of the quote, we refer to those as implicit quotes. Examples of such quotes are given in Figure 1.

Attributing anaphoric and implicit quotes usually requires fine-grained understanding of contextual information such as discerning discussion patterns and long-range information such as linking pronouns to their canonical entity. Accessing information of fictional characters (style, persona, gender) also helps disambiguate challenging examples of attribution. Variations of writing style and

---
[1]Code and data can be found at https://github.com/deezer/character_embeddings_qa

long-range contextual dependencies, ubiquitous in fictional works, add an extra layer of complexity, making attribution of non-explicit quotes a complex task to solve even with recent tools. (Sims and Bamman, 2020; Vishnubhotla et al., 2022)

Some of these shortcomings have been addressed by improving link prediction between pronouns/common nouns and canonical characters (Vishnubhotla et al., 2023) or by exploiting better contextual representations (Cuesta-Lazaro et al., 2022; Su et al., 2023). Instead, we show that combining fictional character representations encoding global stylistic information and topical preferences with contextual information processed by BookNLP, [2], a widely used NLP pipeline designed for literary texts, improves the speaker attribution of anaphoric and implicit quotes. As explicit utterances are often accurately attributed to their speaker (Vishnubhotla et al., 2023), we assume those quotes given, and use them to derive stylistic information of characters.

Among methods to represent characters from the words they utter, previous works (Li et al., 2023; Aggazzotti et al., 2023; Michel et al., 2024) have explored embeddings encoding topical preferences using SBERT (Reimers and Gurevych, 2019) or stylistic information using Universal Authorship Representations (UAR) (Rivera-Soto et al., 2021), a fine-tuned variant of SBERT trained on the Authorship Verification (AV) task with millions of Reddit users. UAR's stylometric embeddings built from character utterances were previously shown to distinguish fictional characters among the same work (Michel et al., 2024) but considering the discrepancies between Reddit and literary discourse, we question the effectiveness of these embeddings to represent characters. To overcome this potential limitation, we build variants of UAR tailored for literary texts that we validate on the AV task with drama and novel characters.

Stylometric character representations computed with our UAR variants significantly improve AV of characters in literary works, while also reaching state-of-the-art quotation attribution accuracy of non-explicit quotes when combined with BookNLP contextual representations. To sum up, our contributions are:

- We construct DramaCV, a corpus of English drama plays, dating from the 15th to 20th century, focused on the AV of fictional characters

within the same work, and train variants of UAR on this dataset, showing that our proposed models significantly improve AV on DramaCV and also generalize better to novels.

- Through an extensive evaluation, we show that BookNLP can be easily extended to improve its current performance on the Project Dialogism Novel Corpus (PDNC) (Vishnubhotla et al., 2022), and reaches state-of-the-art attribution accuracy of non-explicit quotes when trained in combination with global character representations computed from their explicit quotes.

2 Related Work

Quotation Attribution. Datasets of quotation attribution containing fully annotated English novels are rather scarce, and are often limited to one or two books. Elson and McKeown (2010) introduce the CQSC corpus and attribute automatically extracted quotes to named entities and nominals with a supervised mention ranking model. Instead, He et al. (2013) attribute quotes directly to *speakers* with a supervised ranking system using features such as speaker alternation patterns and character-level features (He et al., 2010). Other approaches involve sequence labeling (O'Keefe et al., 2012), Dialogue State Tracking (Cuesta-Lazaro et al., 2022) on English novels, or scoring systems on Chinese novels (Chen et al., 2021). The deterministic sieve-based model of Muzny et al. (2017b) regards quotation attribution as a two-step process: quote-*mention* linking and mention-*speaker* linking. The NLP pipeline dedicated to books, BookNLP, replaced the deterministic sieves with fine-tuned language models. The link between mention and speaker is carried out by BookNLP's coreference resolution model, and the quote-mention linking is predicted by a BERT model, fine-tuned on LitBank's speaker annotations (Sims and Bamman, 2020). Recently, Vishnubhotla et al. (2022) introduce the largest-to-date corpus of quotation attribution, PDNC, and show a similar accuracy score of around 63% for both BookNLP and the sieve-based model. However, better results were obtained later by fine-tuning BookNLP on PDNC and restricting the predicted coreference chains to resolved characters only (Vishnubhotla et al., 2023). SIG (Su et al., 2023) adapted quotation attribution for encoder-decoder models, showing that predict-

ing both the speaker and the addressees of a quote with BART reaches state-of-the-art accuracy on PDNC.

Interestingly, reported BookNLP attribution results on PDNC often approach perfect accuracy for explicit quotes, indicating that most of the research efforts should be put in the attribution of non-explicit quotes. Driven by this insight, we exploit explicit quotes uttered by characters to represent them in a latent space, and are the first to show their potential for quotation attribution when combined with contextual representations.

Fictional Character Embeddings. Understanding fictional characters objectives, personas and style often comes naturally when reading novels. When accessing information from the fictional world and interactions between characters provided throughout the narrative, we infer mental representations of characters that are revised when accessing new information (Gernsbacher et al., 1992; Culpeper, 1996). Previous computational works on this aspect of fictional character understanding have explored representing fictional characters in a latent space, often targeting a specific aspect, such as persona, style or descriptive elements. These computational representations of characters are often used for further analysis of large literary corpora, trying to understand character distinctiveness between authors (Bamman et al., 2014) or narrative comprehension (Brahman et al., 2021). Other works focus on using fictional characters embeddings to improve character-level tasks such as coreference resolution and character linking (Li et al., 2023) or contextual and book-level tasks such as narrative question answering (Inoue et al., 2022).

Here, we propose to represent character from *what they say* and *how they say it*, building literary embeddings encoding style and topical preferences of characters, derived from models trained on the Authorship Verification (AV) of fictional characters within the same work. Most works exploring the AV of fictional characters focus on drama characters because of the availability of large annotated corpora and aim at capturing syntactic, lexical and phonological variations that occur in their direct speech using statistical measures of distinctiveness (Dinu and Uban, 2017; Vishnubhotla et al., 2019; Šeļa et al., 2023).

With the recent introduction of PDNC (Vishnubhotla et al., 2022), a corpus of 22 novels anno-

tated with quotation attribution[3], analysis on fictional characters in novels can be done at a larger scale. Michel et al. (2024) provide a comparative study of PDNC's characters, converting the original PDNC quotation attribution task to the AV of fictional characters and showing that UAR (Rivera-Soto et al., 2021) embeddings derived from a small collection of utterances are able to distinguish characters within the same work. However, Reddit – the target domain of UAR – is inherently different from traditional novels that use conventions on grammar, punctuation and format, motivating the need for a variant of UAR tailed for literary texts.

3 Representing Fictional Characters

In this section, we present our approach to build and evaluate character representations tailored for literary texts, focusing on aspects of style and topical preferences of characters.

To build these representations, we train variants of UAR from scratch on the Authorship Verification (AV) of characters in the literary domain, considering a fictional character as an *author* in the AV terminology. The AV of fictional characters aims at answering the following problem: given two collections of quotes that have been uttered by characters within the same fictional work, predict whether they were spoken by the same character or not. Although closely linked to Authorship Attribution, we note that this framework is different for the following reasons: 1) we do not have access to a list of canonical characters for every literary work and 2) we compare characters within the same story, implying that each quote is inherently influenced by the author's style.

UAR adapts SBERT by directly encoding collections of documents, and is trained to distinguish if two collections of documents have been written by the same author or not. Although the Reddit version of UAR appears to perform well on PDNC (Michel et al., 2024), which is on-par with its zero-shot domain transfer abilities, we question its suitability for literary analysis and propose to train instead a domain-specific variant of UAR tailored for literary texts.

Only a few datasets containing annotations of the speaker of an utterance of direct speech in full novels exist, as annotating entire works is costly. Instead, prior works have considered using movie

---

[3]6 additional novels were released a year later, totaling 28 annotated novels.

	Split	Segments	Utterances	Queries	Targets/Query (avg)
Scene	Train	1507	263270	5392	5.0
	Val	240	50670	1557	8.8
	Test	203	41830	1319	8.7
Play	Train	226	449407	4109	90.7
	Val	30	63934	917	55.1
	Test	31	74738	1214	108.5
PDNC	Explicit	-	6303	562	11.2

Table 1: Summary statistics of the Scene and Play instances of DramaCV by split. A segment in the Scene split is a single scene or act in a play, while a segment in the Play split is the entire play. For each segment, the targets are collections of utterances uttered by all participating characters in the segment. The Explicit split of PDNC (Michel et al., 2024) that we use to test generalization is also displayed for comparison.

(Azab et al., 2019; Sang et al., 2022; Li et al., 2023) or drama scripts (Vishnubhotla et al., 2019; Fischer et al., 2019) as they come naturally annotated with the speaker of each utterance. We follow this approach and propose DramaCV (Drama Character Verification), a dataset of English drama plays focused on the AV of fictional characters. We chose to use drama plays as they are mostly similar to novels in terms of writing conventions, and because we were able to extract a number of characters and utterances an order of magnitude above the available datasets of annotated novels. The rest of this section evaluates our variants of UAR trained on DramaCV on the test set of DramaCV and on fictional characters participating in the 28 PDNC novels.

3.1 DramaCV

We extract 499 plays dating from the 15th to 20th century from Project Gutenberg[4] with the GutenTag software (Brooke et al., 2015) that we automatically parsed to create a character list and to attribute each line to its speaker. Then, we construct two instances of DramaCV:

Scene. We split each play in scenes, a small segment unit of drama that is supposed to contain actions occurring at a specific time and place with the same characters. If a play has no `<scene>` tag, we instead split it in acts, with the `<act>` tag. Acts are larger segment units, composed of multiple scenes. For this split, we only consider plays that have at least one of these tags. Although scenes usually contain a few characters, they represent a challenge for CV models as participating characters are akin to discuss a similar topic. Thus, stylistic cues of characters are likely to be useful to distinguish char-

acters within the same scene.

Play. We do not segment play and use all character lines in a play. Compared to the scene segment, the number of candidate characters is higher, and discussions could include various topics. Hence, topical preferences of characters might be an additional useful feature to distinguish characters.

After parsing, a total of 169 plays for the Scene split, and 287 plays for the Play split remained. For both splits, we use 80% of the plays for training, 10% for validation and 10% for test. Following the AV setup, we build queries and targets using collections of character utterances. In this setup, the objective is to recognize if a query and a target have been uttered by the same character or not. We construct queries in the validation and test splits by randomly sampling half of the utterances of a character, and we use the remaining half as the target. The train splits only consider characters that uttered at least 16 lines, such that we can build queries and targets with at least 8 utterances. More details on the construction of queries and targets can be found in Appendix A. Summary statistics for both instances of DramaCV are presented in Table 1.

3.2 Literary UAR

We train UAR from scratch using its public implementation on DramaCV. For both instances, we use the pre-trained `all-distilroberta-v1` as the base encoder, and employ a learning rate of 2e-5 for 20 epochs. We set the dimension of the model to $d = 512$ and use a maximum sequence length of 64 tokens for all utterances. During training, we build queries and targets by randomly sampling 8 utterances to represent characters. A batch size of 1 play is used when training on the Play split, and

---

[4] https://gutenberg.org/

8 scenes are used when training on the Scene split. During inference, we do not restrict the number of utterances in a collection to reflect situations where we have access to only less than 8 utterances, or to more than 8 utterances.

A supervised contrastive objective (Khosla et al., 2020) is used where in-batch negatives considered are characters within the same segment unit. We motivate this choice for each dataset instance:

Scene. In a scene, characters are likely to discuss similar topics. By restricting the objective to characters within a scene, we force the model to rely on stylistic cues to distinguish lines of characters.

Play. We want to avoid capturing the writing style of authors to distinguish between collections of character lines. Thus, we restrict the set of in-batch negatives to characters within the same work.

We compare our variants of UAR, UAR_{Scene}[5] and UAR_{Play}[6] against their Reddit counterpart, UAR_{Reddit} and SBERT. Following previous work (Aggazzotti et al., 2023; Michel et al., 2024), we create SBERT character embeddings by encoding independently each utterance in a feature d-dimensional vector[7], followed by a coordinate-wise mean of all utterance vectors. Instead, UAR directly encodes a collection of utterances into a d-dimensional vector.

Given a collection of a character utterances (query) and collections of all characters occurring in the same segment (targets, that include a target uttered by the same character as the query), we evaluate the ability of these models to yield a higher cosine-similarity to the target uttered by the same character than to targets uttered by other characters using Area Under the Receiver Operating Characteristic curve (AUC). We use this evaluation setup on fictional characters for both test-sets of DramaCV and for the 28 novels of PDNC on the Explicit split of Michel et al. (2024). More details on the evaluation can be found in Appendix A.

3.3 Authorship Verification Results

We report AUC for both test splits of DramaCV and the Explicit split of PDNC in Table 2. Interestingly, UAR_{Reddit} performs worse on both splits compared to SBERT and in-domain UARs. UAR_{Scene} increases AUC on its training domain up to 10 points compared to SBERT, indicating that the model is able to capture stylistic cues specific

---

[5] https://huggingface.co/gasmichel/UAR_Scene
[6] https://huggingface.co/gasmichel/UAR_Play
[7] We use all-mpnet-base-v2.

	DramaCV		PDNC - Exp	
	Scenes	**Plays**	**CC**	**CQ**
SBERT	72.0 (5)	78.8 (10)	63.8 (16)	54.4 (5)
UAR_{Reddit}	68.6 (7)	74.5 (11)	81.1 (9)	54.1 (4)
UAR_{Scene}	**82.3** (6)	-	81.3 (10)	**58.9** (4)
UAR_{Play}	-	**84.2** (9)	**86.5** (8)	58.5 (5)

Table 2: Average AUC (%) on test plays in DramaCV and on novels in the Explicit split of PDNC with standard deviations between parentheses.

to drama that help at distinguishing utterances of characters discussing similar topics. The increase of UAR_{Play} is slightly lower, but the high AUC indicates that it succeeds at distinguishing characters within the same play. The high standard deviation across plays suggests that task difficulty is not distributed uniformly among these literary works. Among best-performing plays, we found works by canonical authors such as Shakespeare's *Hamlet*, Shaw's *John Bull's Other Island* or Ben Jonson's *The Alchemist* (see complete list in Appendix B).

Looking at generalization results of PDNC, UAR_{Reddit} performs better than SBERT, and almost equals UAR_{Scene} in the Character-Character setup (CC), which compares similarity of character representations. We find significant improvement with UAR_{Play} over the baselines, suggesting that training UAR on drama allows better domain adaptation to literary novels. We also remark that this adaptation is data-efficient, as training UAR_{Scene} with around 5000 drama characters is enough to match the performance of UAR_{Reddit} that has been trained on millions of different users.

The advantages of domain adaptation can also be seen in the Character-Quote (CQ) evaluation setup that aims at attributing utterances to their speaker by comparing the similarity between character and quote embeddings computed with the same model. Here, the two variants of UAR trained on drama show a significant increase in AUC over the baselines, indicating that character representations built with our proposed UARs are more similar to embeddings of their other quotes. However, AUC scores for the CQ experiment are low, showing that the task of attributing quotes requires more than only comparing similarities of character and quote embeddings.
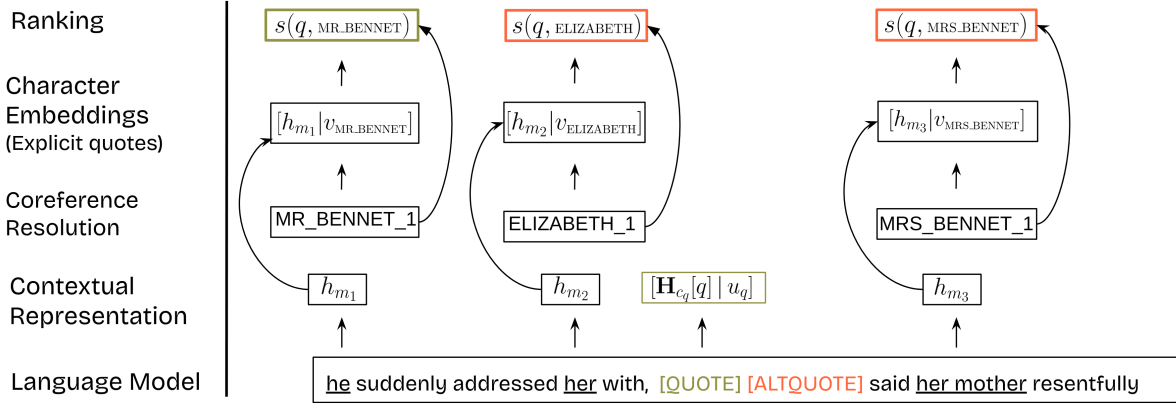
Ranking

$$s(q, \text{MR\_BENNET}) \quad s(q, \text{ELIZABETH}) \quad s(q, \text{MRS\_BENNET})$$

Character Embeddings (Explicit quotes)

$$[h_{m_1} | v_{\text{MR\_BENNET}}] \quad [h_{m_2} | v_{\text{ELIZABETH}}] \quad [h_{m_3} | v_{\text{MRS\_BENNET}}]$$

Coreference Resolution

$$\text{MR\_BENNET\_1} \quad \text{ELIZABETH\_1} \quad \text{MRS\_BENNET\_1}$$

Contextual Representation

$$h_{m_1} \quad h_{m_2} \quad [\mathbf{H}_{c_q}[q] \,|\, u_q] \quad h_{m_3}$$

Language Model

he suddenly addressed her with, [QUOTE] [ALTQUOTE] said her mother resentfully

Figure 2: Detailed process of BookNLP's quotation attribution system. We inject extra-information in the form of character embeddings computed from explicit quotes $v_{\text{entity}(m)}$ in the fourth step, and quote embeddings u_q in the second step.

4 Quotation attribution

Motivated by the ability of UAR to encode character utterances in a representation space that contains useful, though insufficient, information for quotation attribution, we evaluate in the next section the impact of combining these global character embeddings with BookNLP's contextual information. BookNLP is a Natural Language Processing pipeline dedicated to English books, popular among practitioners of Digital Humanities.

4.1 BookNLP

BookNLP's pipeline includes, among others, Part-of-speech tagging, Named Entity Recognition, Coreference Resolution and quotation attribution. All components of the pipeline have been trained on LitBank (Bamman et al., 2019, 2020; Sims and Bamman, 2020), a corpus containing 100 chapters from public domain books. Since its quotation attribution model is not part of a publication, we will review its details in the following paragraphs.

BookNLP carries out quotation attribution in four independent steps: 1) tokenization and named entity recognition, 2) character name clustering, 3) pronominal coreference resolution, and 4) quote-mention linking. The first three steps are dedicated to extract named, nominal and pronominal mentions along with the predicted character they refer to. Using extracted mentions in the surrounding context of a quote, the final step, quote-to-mention linking, aims at predicting which mention is the one that refers to the true speaker of the quote. In other words, it finds the information given by the narrator about who is speaking among the narrative parts in the vicinity of quotes. It is achieved by a fine-tuned BERT model (Devlin et al., 2019). We give further details of the quote-mention linking step below.

Let $D = (t_1, \ldots t_n)$ be a tokenized document and $q = (t_i, \ldots, t_j)$ be a character utterance in this document, starting at token i and ending at token j. A contextual segment is computed for each quote, $c_q = [\, c_q^{left} \,|\, q \,|\, c_q^{right} \,]$, where c_q^{left} and c_q^{right} are left and right contextual information of length w tokens and $[\,|\,]$ denotes concatenation. Additionally, the full quote q is replaced by a special token [QUOTE] and all other quotes occurring in the contextual segment are replaced by a special token [ALTQUOTE], such that the contextual segment mainly reflect narrative parts. Each segment c_q, containing masked quotes and narrative parts, is fed to BERT, producing a contextual representation $\mathbf{H}_{c_q}$. Then an unary compatibility score between the (masked) quote token q and a mention-span $m = (t_{m_s}, \ldots, t_{m_e})$ occurring in the segment is calculated as follows:

$$h_m = \frac{1}{|m|} \sum_{i=m_s}^{m_e} \mathbf{H}_{c_q}[i] \qquad (1)$$

$$s(q, m) = \phi([\mathbf{H}_{c_q}[q] \,|\, h_m]) \qquad (2)$$

where $\mathbf{H}_{c_q}[i]$ is the BERT representation of token i while $|m|$ is the token length of mention m and ϕ is a feed-forward neural network with one hidden layer and a tanh activation. Given a set of candidates $Y(q) = (m_1, \ldots, m_k)$, the model is trained to optimize the log-likelihood of mentions referring to the true speaker of q. Figure 2 summarizes this process.

	Overall	Non-Explicit	Explicit	Anaphoric	Implicit
SIG	72^+	70.0^+	-	-	-
ChatGPT	71^+	70.0^+	-	-	
Unanswerable (%)	14.1 (4.4)	20.9 (6.7)	0	24.5 (8.1)	19.8 (8.6)
BookNLP+$_{reimp}$	78.5 (4.0)	68.9 (4.4)	98.6 (1.2)	70.2 (7.0)	66.4 (5.7)
+ SBERT	$79.2^{\star\star}$ (3.8)	$70.0^\star$ (4.5)	98.3 (1.1)	70.8 (7.3)	$68.1^\star$ (5.5)
+ UAR$_{Reddit}$	$80.1^{\star\star}$ (3.9)	$71.1^{\star\star}$ (4.7)	98.7 (1.1)	71.5 (7.9)	$69.6^\star$ (5.3)
+ UAR$_{Play}$	$80.0^\star$ (3.8)	$70.9^\star$ (4.4)	98.7 (1.0)	71.1 (7.9)	$69.5^\star$ (5.7)
+ UAR$_{Scene}$	$\mathbf{80.2}^\star$ (3.8)	$\mathbf{71.2}^\star$ (4.3)	98.7 (1.0)	$\mathbf{71.7}^{\star\star}$ (8.1)	$69.6^\star$ (5.4)

Table 3: Cross validation accuracy (%) of quotation attribution on the PDNC dataset. Standard deviations in parentheses are calculated across splits. We take the reported accuracy of SIG and ChatGPT (accuracies for Implicit and Anaphoric quotes are not reported) from (Su et al., 2023) (+). Statistical significance against BookNLP+$_{reimp}$ from paired t-test is denoted by $\star$ (5%) and $\star\star$ (10%).

The default window size is $w = 50$ tokens, and the hidden layer dimension is 100. During training and inference, only a maximum of $k = 10$ mention candidates are considered.

In our experiments, we modify these default parameters by increasing the narrative window size to $w = 100$ tokens, and using a relu activation and a hidden dimension of $d = 512$ in the feed-forward neural network. We also replace BERT by SpanBERT-large (Joshi et al., 2020), computing mention representations by concatenating Span-BERT embeddings of the first and last tokens. At inference, we do not truncate the number of candidates and instead use all candidate mentions available in the context window.

4.2 Integrating Character Embeddings

Let $E(c) = (q_i, \ldots, q_j)$ be the collection of all explicit quotes uttered by character c. We embed this collection of quotes using one of the model described in Section 3.2, yielding a vector representation v_c for each character. If a character has not been explicitly quoted, we set its vector representation to zeros. Additionally, we embed each target quote q independently with the same model to get a vector representation u_q. We modify Equation 2 to combine BookNLP's contextual representation with these character and quote representations:

$$s(q, m) = \phi([\mathbf{H}_{c_q}[q] \mid h_m \mid v_{\text{entity}(m)} \mid u_q]) \quad (3)$$

where entity(m) is the character predicted by the coreference resolution step for mention m.

Equation 3 allows BookNLP to rely on external information by injecting global character information and compatibility between quote q and candidate character c.

4.3 Experimental Setup

Following Su et al. (2023), we use the first version of PDNC, which contains 22 annotated novels. We focus on *major* and *intermediate* characters, who are characters that uttered at least 10 quotes in the novel. This allows to avoid the long tail of minor characters and to compare our results to the current state-of-the-art, SIG and ChatGPT (*gpt-3.5-turbo-0613*).

We base our implementation on BookNLP+ (Vishnubhotla et al., 2023), that restricts coreference predictions to chains that can be successfully resolved to a character entity from the annotated PDNC character lists. We follow the same evaluation protocol, using their provided cross-validation split at the novel level, reporting the average accuracy over each split.

We train our implementation of BookNLP+, BookNLP+$_{reimp}$ with the modifications mentioned in Section 4.1, and add character embeddings encoded by different models using Equation 3. We use AdamW optimizer with a learning rate of 5e-6 and train each model for 20 epochs.

4.4 Results

Our main results are displayed in Table 3. Our implementation of BookNLP+ reaches 68.9% accuracy on non-explicit quotes, compared to the 53% reported in Vishnubhotla et al. (2023). This large increase is mostly due to the replacement of BERT-base with SpanBERT-large. Over all quotes, BookNLP+$_{reimp}$ largely outperforms SIG and ChatGPT.

Combining BookNLP+ with SBERT character embeddings matches the accuracy of SIG and ChatGPT on non-explicit quotes. The gains are larger for all variants of UAR, reaching 71.2% ac-
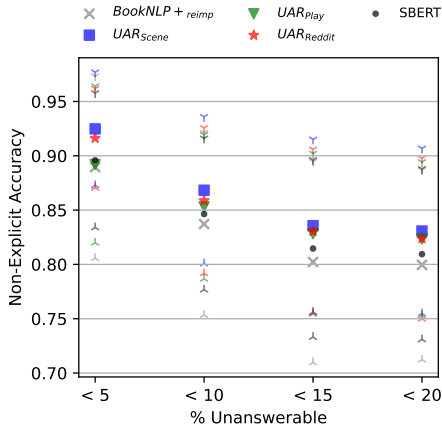
Figure 3: Accuracy results on non-explicit quotes averaged over subsets of novels that have a low percentage of unanswerable quotes, with standard deviations.

	Δ **Non-Exp**	Δ **Ana**	Δ **Imp**
SBERT	-0.0	-0.0	-0.1
UAR_{Reddit}	-0.1	-0.0	-0.2
UAR_{Play}	$-0.2^{\star\star}$	-0.1	$-0.3^{\star}$
UAR_{Scene}	$+0.1$	$+0.1$	$+0.1$

Table 4: Original accuracy (Table 3) variation when replacing gold explicit attributions with attributions predicted by BookNLP+$_{reimp}$. Statistical significance against original accuracy from paired t-test is indicated with $^{\star}(5\%)$ and $^{\star\star}(10\%)$

curacy on non-explicit quotes with our proposed UAR_{Scene}. As expected, we found that these gains are largely due to the improved attribution performance of implicit quotes, increasing the accuracy by more than 3 points with all UAR models.

Interestingly, although our proposed variants of UAR tailored for literary texts are better at the Authorship Verification of fictional characters, we only see a small – though statistically significant – increase between UAR_{Scene} and its Reddit counterpart on non-explicit attribution accuracy. The reasons could be twofold. First, BookNLP's system only focuses on finding the right narrative information that helps identifying the speaker of a quote. However, our embeddings mostly reveal stylistic cues of characters, which are not encoded by BookNLP. Therefore, our implementation provides external character information that might not align perfectly with BookNLP's contextual decisions, limiting their potential combination. Second, we add character information based on noisy coreference chains, which might hinder the model's performance. We believe further research should be conducted into finding better combinations between character embeddings and BookNLP.

The high standard deviations across splits reveals that the task is harder for some novels than others. Looking at the percentage of unanswerable quotes (quotes that do not have a candidate mention referring to their speaker in their context window) provides a reason for such high variations: it ranges from 0 to 56% per novel. Figure 3 shows the attribution accuracy of non-explicit quotes when only considering novels that have a low percentage of

unanswerable quotes. Only 4 novels have a percentage lower than 5%, but 17 out of the 22 have a percentage lower than 20%. As expected, we see higher performance for all models, and the gain from UAR character embeddings is more clear.

Let us note that UAR_{Scene} is consistently better than UAR_{Play}, though the latter is better at distinguishing PDNC characters in the CC experiment. Restricting UAR to encode stylistic information by using collections of quotes with similar topic seems to generalize better to quotation attribution.

To construct character embeddings, we relied on gold-labeled attributions of explicit quotes. We report in Table 4 the accuracy variation when replacing the gold attributions with speaker predictions from BookNLP+$_{reimp}$. Only UAR_{Play} shows a statistically significant decrease in accuracy, although this variation remains very small. This validates our intuition that explicit quotes can be extracted in a first step and that the eventual noise contained in the resulting character embeddings does not harm their performance for quotation attribution.

Finally, we present quotation attribution results for 6 annotated novels from the second release of PDNC in Table 5. We use the model trained on the first cross-validation split to predict attributions in each of these novels. The impact of using literary character embeddings is more clear: both of our proposed variants of UAR show increased attribution accuracy against their Reddit counterpart. Besides, the increase in accuracy over non-explicit quotes is again mostly attributed to an increase in implicit speaker identification.

5 Conclusion

In this work, we evaluated fictional character embeddings constructed with the explicit quotes they utter, showing that they improve quotation attribution accuracy of non-explicit quotes on PDNC when used in combination with BookNLP's contex-

	Non-Exp	Ana	Imp
Unanswerable (%)	20	21	18
BookNLP+$_{reimp}$	69.6	72.3	67.6
+ SBERT	69.3	71.6	68.0
+ UAR$_{Reddit}$	70.5**	72.4	69.4*
+ UAR$_{Play}$	71.1*	72.8**	**70.1***
+ UAR$_{Scene}$	**71.2***	**73.1**	70.0*

Table 5: Average quotation attribution accuracy (%) over the 6 additional PDNC novels. Statistical significance from paired t-test against BookNLP+$_{reimp}$ is denoted by * (5%) and ** (10%).

tual representations. We created DramaCV, a corpus of English drama plays focused on the Authorship Verification of fictional characters and trained two variants of UAR on this dataset. Our proposed models show better performance when distinguishing fictional characters in both DramaCV and PDNC. Character embeddings derived from these models are further used for quotation attribution on PDNC, reaching state-of-the-art accuracy when combined with BookNLP+. The impact of using literary character embeddings against embeddings derived from a model trained on Reddit is small, though statistically significant, suggesting superiority of in-domain character representations for quotation attribution.

We proposed a straightforward way to combine fictional character embeddings with contextual representations, using BookNLP as an example. Our approach can be adapted to other models: cross-attention over character embeddings could be used for encoder-decoder models, while modality alignment (Zhang et al., 2023) is a solution for decoder-only models. We leave potential improvements to the combination of fictional character embeddings and contextual representations to future work.

We found that BookNLP's attribution accuracy is not distributed uniformly across novels, mainly caused by noisy coreference resolution, resulting in many quotes not having contextual candidate mentions that refer to their true speaker. We believe that including our character embeddings in the coreference resolution model as in Li et al. (2023) might also improve its performance and thus increase overall attribution accuracy. Another potential solution could include unmasking the closest quotes in the vicinity of a target quote that are likely to contain addressee information. We leave better combination of fictional character embeddings with BookNLP for future exploration.

6 Limitations

In this work, we focused on integrating fictional character embeddings in encoder-only models. We have not tested with other types of models such as encoder-decoder and decoder only, but we believe that integrating our embeddings in those frameworks is possible and is likely to improve their performance on the quotation attribution task. Our first set of experiments corroborate this assumption, on the usefulness of our proposed character embeddings in distinguishing fictional characters.

We also proposed models that can be used to create character representations. These models are trained on a corpus of drama plays, which spans almost 400 years of work. It is unclear how the variation of writing styles that have occurred throughout these centuries affects the information contained in these embeddings, and how they can generalize to new literary styles. This remark also extends to our results on quotation attribution. We presented results on a corpus containing novels from the 19th to early 20th century ; which is closer in terms of writing style to DramaCV than newer novels published in the 21th century. It is thus unclear how our results generalize to these newer novels, but we strongly believe that accurate character embeddings improve quotation attribution regardless of the writing style of the author.

We only focused on representations encoding stylistic cues and topical preferences and showed that they improve quotation attribution. However, this type of character embedding might be incomplete and other aspects of characters can be used in addition to our proposed embeddings (persona, gender, objective, or social status). This might allow to build accurate representations of characters that are likely to improve even more quotation attribution accuracy.

7 Acknowledgements

References

Cristina Aggazzotti, Nicholas Andrews, and Elizabeth Allyn Smith. 2023. Can authorship attribution models distinguish speakers in speech transcripts? Preprint, arXiv:2311.07564.

Mahmoud Azab, Noriyuki Kojima, Jia Deng, and Rada Mihalcea. 2019. Representing movie characters in

dialogues. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 99–109, Hong Kong, China. Association for Computational Linguistics.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

David Bamman, Sejal Popat, and Sheng Shen. 2019. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2138–2144, Minneapolis, Minnesota. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A Bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379, Baltimore, Maryland. Association for Computational Linguistics.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. "let your characters tell their story": A dataset for character-centric narrative understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1734–1752, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Julian Brooke, Adam Hammond, and Graeme Hirst. 2015. GutenTag: an NLP-driven tool for digital humanities research in the Project Gutenberg corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 42–47, Denver, Colorado, USA. Association for Computational Linguistics.

Yue Chen, Zhen-Hua Ling, and Qing-Feng Liu. 2021. A Neural-Network-Based Approach to Identifying Speakers in Novels. In *Proc. Interspeech 2021*, pages 4114–4118.

Carolina Cuesta-Lazaro, Animesh Prasad, and Trevor Wood. 2022. What does the sea say to the shore? a BERT based DST style approach for speaker to dialogue attribution in novels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5820–5829, Dublin, Ireland. Association for Computational Linguistics.

Jonathan Culpeper. 1996. Inferring character from texts: Attribution theory and foregrounding theory. *Poetics*, 23(5):335–361.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liviu P. Dinu and Ana Sabina Uban. 2017. Finding a character's voice: Stylome classification on literary characters. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 78–82, Vancouver, Canada. Association for Computational Linguistics.

David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden. Association for Computational Linguistics.

David Elson and Kathleen McKeown. 2010. Automatic attribution of quoted speech in literary narrative. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1):1013–1019.

Frank Fischer, Ingo Börner, Mathias Göbel, Angelika Hechtl, Christopher Kittel, Carsten Milling, and Peer Trilcke. 2019. Programmable Corpora: Introducing DraCor, an Infrastructure for the Research on European Drama. In *Proceedings of DH2019: "Complexities", Utrecht, July 9–12, 2019*. Utrecht University.

Morton Ann Gernsbacher, H Hill Goldsmith, and Rachel RW Robertson. 1992. Do readers mentally represent characters' emotional states? *Cognition & Emotion*, 6(2):89–111.

Hua He, Denilson Barbosa, and Grzegorz Kondrak. 2013. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1312–1320, Sofia, Bulgaria. Association for Computational Linguistics.

Hua He, Greg Kondrak, and Denilson Barbosa. 2010. The actor-topic model for extracting social networks in literary narrative.

Naoya Inoue, Charuta Pethe, Allen Kim, and Steven Skiena. 2022. Learning and evaluating character representations in novels. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1008–1019, Dublin, Ireland. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron

Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys*, 52(5):1–40.

Dawei Li, Hengyuan Zhang, Yanran Li, and Shiping Yang. 2023. Multi-level contrastive learning for script-based character understanding. *Preprint*, arXiv:2310.13231.

Gaspard Michel, Elena Epure, Romain Hennequin, and Christophe Cerisara. 2024. Distinguishing fictional voices: a study of authorship verification models for quotation attribution. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 160–171, St. Julians, Malta. Association for Computational Linguistics.

Grace Muzny, Mark Algee-Hewitt, and Dan Jurafsky. 2017a. Dialogism in the novel: A computational model of the dialogic nature of narration and quotations. *Digital Scholarship in the Humanities*, 32:ii31–ii52.

Grace Muzny, Michael Fang, Angel Chang, and Dan Jurafsky. 2017b. A two-stage sieve approach for quote attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 460–470, Valencia, Spain. Association for Computational Linguistics.

Timothy O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 790–799, Jeju Island, Korea. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yisi Sang, Xiangyang Mou, Mo Yu, Shunyu Yao, Jing Li, and Jeffrey Stanton. 2022. TVShowGuess: Character comprehension in stories as speaker guessing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4267–4287, Seattle, United States. Association for Computational Linguistics.

Matthew Sims and David Bamman. 2020. Measuring information propagation in literary social networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 642–652, Online. Association for Computational Linguistics.

Zhenlin Su, Liyan Xu, Jin Xu, Jiangnan Li, and Mingdu Huangfu. 2023. Sig: Speaker identification in literature via prompt-based generation. *Preprint*, arXiv:2312.14590.

Krishnapriya Vishnubhotla, Adam Hammond, and Graeme Hirst. 2019. Are fictional voices distinguishable? classifying character voices in modern drama. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 29–34, Minneapolis, USA. Association for Computational Linguistics.

Krishnapriya Vishnubhotla, Adam Hammond, and Graeme Hirst. 2022. The project dialogism novel corpus: A dataset for quotation attribution in literary texts. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5838–5848, Marseille, France. European Language Resources Association.

Krishnapriya Vishnubhotla, Frank Rudzicz, Graeme Hirst, and Adam Hammond. 2023. Improving automatic quotation attribution in literary novels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 737–746, Toronto, Canada. Association for Computational Linguistics.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773, Singapore. Association for Computational Linguistics.

Artjoms Šela, Ben Nagy, Joanna Byszuk, Laura Hernández-Lorenzo, Botond Szemes, and Maciej Eder. 2023. From stage to page: language independent bootstrap measures of distinctiveness in fictional speech. *Preprint*, arXiv:2301.05659.

A Details on the Construction and Evaluation of DramaCV

Each play collected with GutenTag is encoded in a 'XML' file, containing various tags such as the

<speaker> tag that indicates the speaker of a line. We parse each XML file, removing plays written by non English-speaking authors, and attribute each line to its speaker using the <speaker> tag.

For the Scene split, we automatically check if a play contains a <scene> or <act> tag. We segment plays by default with the <scene> tag, and use the <act> tag if it does not contain a scene tag. We removed plays that did not contain any of these tags. For the Play split, we kept all plays written by English-speaking authors.

In the Authorship Verification terminology, queries and targets are collections of documents written by either the same or different authors. Given a query written by author A and multiple targets written by either the same or different authors, the objective of AV is to recognize all targets written by author A. Details of queries and target construction and our evaluation setup are provided below:

Queries/Targets. Given characters $(c_1, \ldots, c_n)$ occurring in a single segment (a scene/act for the Scene split, or the entire play for the Play split), we randomly sample half of their utterances to construct queries $(q_{c_1}, \ldots, q_{c_n})$ and use the remaining half to construct targets $(t_{c_1}, \ldots, t_{c_n})$. For the train split, we instead randomly sample 8 utterances to build queries and 8 different utterances to build targets.

DramaCV Evaluation. Using any model $\mathcal{M}$ that is able to encode collections of documents in a feature vector, we encode each query and target independently. Then, we compare the cosine similarity of query vectors with targets: $similarity(\mathcal{M}(q), \mathcal{M}(t)) \ \forall q \in q_{c_1}, \ldots, q_{c_n}$ and $\forall t \in t_{c_1}, \ldots, t_{c_n}$. For a query q_c uttered by character c, we calculate an AUC score over the resulting cosine similarities with targets, where a label 1 is assigned to t_c and 0 to other targets. This AUC represents the ability of $\mathcal{M}$ to assign an higher cosine similarities between queries and targets written by the same author. We average this AUC score over each query in a single segment to obtain an AUC score for this particular segment.

PDNC Evaluation. We use the evaluation setup of Michel et al. (2024) to evaluate our models on the characters of PDNC. This setup proposes two types of evaluation: Character-Character (CC) and Character-Quotes (CQ).

The CC evaluation is similar to the one described above, but the queries are constructed using only the explicit quotes of a character in a single chapter, and targets are constructed using every other utterance in the other chapters. The AUC computed scores how a model is able to assign a higher cosine-similarity between a collection of a few explicit quotes of character, and its other utterances in the novel.

In the CQ setup, queries are similar to CC, but a target is defined as a single utterance in the held-out chapters, without access to contextual information. Therefore, the AUC computed is scoring how similar the collection of a few explicit quotes of character is to its other individual quotes. This task is inherently harder than the CC setup, as an individual quote might be very different from other quotes (e.g "Hey, how are you Lizzy?").

B Performance on all Test Plays

We report performance of UAR$_{Scene}$ and UAR$_{Play}$ on their respective test split in Table 6 and Table 7.

C Computing Information

We used a 32-core Intel Xeon Gold 6244 CPU @ 3.60GHz CPU with 128GB RAM equipped with 3 RTX A5000 GPUs with 24GB RAM each and a single Nvidia A40 with 45GB RAM. We trained UAR$_{Scene}$ and UAR$_{Play}$ with the Nvidia A40, requiring gradient-checkpointing to fit everything in the GPU RAM. The training took around 1 hour for both models. We use a single RTX A5000 for training BookNLP+$_{reimp}$ and its variant with character embeddings. Training on a single cross-validation split took around 6 hours for each of the proposed methods. Inference times are shorter, where quotation attribution on a full novel can be done in a few minutes only.

Play	Author	AUC
Justice	John Galsworthy	95.6
The Alchemist	Ben Jonson	91.8
Hamlet	William Shakespeare	86.5
Plays : Third Series	John Galsworthy	86.3
The works of John Dryden	John Dryden	85.2
The Life of Timon of Athens	William Shakespeare	84.2
King Henry the Fifth	William Shakespeare	83.7
Night Must Fall : a Play in Three Acts	Emlyn Williams	83.7
The Spanish Tragedy	Thomas Kyd	83.2
The works of John Dryden	John Dryden	83.0
The Scarlet Stigma: A Drama in Four Acts	Nathaniel Hawthorne	82.4
King John	William Shakespeare	81.8
The Works of Charles and Mary Lamb	Mary Lamb	80.8
The Scornful Lady	John Fletcher	80.8
Married Life A Comedy, in Three Acts	John Baldwin Buckstone	80.0
The Flutter of the Goldleaf; and Other Plays	Frederick Peterson	77.5
Single Life A Comedy, in Three Acts	John Baldwin Buckstone	75.9
The Countess Cathleen	W. B. (William Butler) Yeats	75.6
De Turkey and De Law: A Comedy in Three Acts	Zora Neale Hurston	66.4

Table 6: AUC per novel on the Scene test split of DramaCV.

Play	Author	AUC
Bride Roses	William Dean Howells	100.0
Fanny's First Play	Bernard Shaw	100.0
Five Little Plays	Alfred Sutro	98.6
John Bull's Other Island	Bernard Shaw	95.4
Mrs. Dot: A Farce	William Somerset Maugham	94.5
The Earl of Essex: A Tragedy, in Five Acts	Henry Jones	94.4
The Way of the World	William Congreve	92.7
Plays : Fifth Series	John Galsworthy	92.5
Representative Plays by American Dramatists: 1856-1911: The New York Idea	Langdon Elwyn Mitchell	91.9
Plays: Lady Frederick, The Explorer, A Man of Honour	William Somerset Maugham	90.3
Great Catherine	Bernard Shaw	89.3
Hypolympia Or, The Gods in the Island, an Ironic Fantasy	Edmund Gosse	87.1
The works of John Dryden	John Dryden	87.0
The Grecian Daughter	Arthur Murphy	85.5
Dramatic Technique	George Pierce Baker	81.6
Lyre and Lancet: A Story in Scenes	F. Anstey	80.2
Beaumont and Fletcher's Works, Vol. 5	Francis Beaumont	79.0
King Henry VI, Part 3	William Shakespeare	78.1
Plays and Lyrics	Cale Young Rice	78.0
The Easiest Way Representative Plays by American Dramatists: 1856-1911	Eugene Walter	77.6
Charles Di Tocca: A Tragedy	Cale Young Rice	77.3
St. Patrick's day, or, the scheming lieutenant : a farce in one act	Richard Brinsley Sheridan	76.7
Deirdre of the Sorrows	John Millington Synge	76.3
Ambrose Gwinett or, a sea-side story : a melo-drama, in three acts	Douglas William Jerrold	75.8
Henry V	William Shakespeare	75.3
Box and Cox: A Romance of Real Life in One Act.	John Maddison Morton	75.0
The Recruiting Officer	George Farquhar	74.7
Yolanda of Cyprus	Cale Young Rice	74.3

Table 7: AUC per novel on the Play test split of DramaCV.