What does the sea say to the shore?
A BERT based DST style approach for speaker to dialogue attribution in novels

Carolina Cuesta-Lazaro[†*] **Animesh Prasad**[‡] **Trevor Wood**[‡]
[†] Institute for Computational Cosmology
Durham University, UK
carolina.cuesta-lazaro@durham.ac.uk
[‡]Alexa AI
Amazon, UK
{animpras, trevowoo}@amazon.com

Abstract

We present a complete pipeline to extract characters in a novel and link them to their direct-speech utterances. Our model is divided into three independent components: extracting direct-speech, compiling a list of characters, and attributing those characters to their utterances. Although we find that existing systems can perform the first two tasks accurately, attributing characters to direct speech is a challenging problem due to the narrator's lack of explicit character mentions, and the frequent use of nominal and pronominal coreference when such explicit mentions are made. We adapt the progress made on Dialogue State Tracking to tackle a new problem: attributing speakers to dialogues. This is the first application of deep learning to speaker attribution, and it shows that is possible to overcome the need for the hand-crafted features and rules used in the past. Our full pipeline improves the performance of state-of-the-art models by a relative 50% in F1-score.

1 Introduction

Natural Language Processing has enabled a quantitative improvement in the humanities, by allowing for large-scale statistical measurements to be taken over hundreds of thousands of books compared to the order of tenths a human could analyse in a much longer time span. Some examples of large-scale literary analyses include studies on characters and their descriptions within the novel, mostly focused on gender differences (Underwood et al., 2018; Kraicer and Piper, 2019), and studies on character's relations by extracting social networks from novels (Labatut and Bost, 2019; Jayannavar et al., 2015).

Most of these studies demand special attention to dialogues, being a major part of character expression and interaction with other characters. Dialogues play an instrumental role in plot development, frequently encompassing focal plot moments, especially in fiction – which is also the focus for this study. Here we aim to identify direct-speech utterances that form part of dialogues and associate them with the speaking characters. Such information is not only useful to enable large-scale sociotemporal studies but also crucial to many downstream challenging tasks like narrative understanding (Iyyer et al., 2016) and summarising (Ladhak et al., 2020). Further, the high-quality dialogue-character association is pertinent for generating engaging text-to-speech for novels with distinct voice profiles for characters.

In the past, models that link direct speech to characters have been dominated by predefined rules (Muzny et al., 2017) or hand-crafted features (He et al., 2013). When evaluating these models the authors also presumed that a character list, together with the character's aliases, has been precompiled and that direct-speech text has been extracted. Although extensions to these models that extract speaking characters in a fully automated manner exist, it is unclear what impact does the automation of the aforementioned steps has on the final performance of the model. Moreover, the models have only been tested against a small dataset of three books from the same time period.

These are the two questions that we aim to answer in this paper: i) how can we build flexible models that can generalise and improve with increasing dataset sizes?, and ii) what is the impact of errors propagating from each component of the pipeline, and thus where should we focus future efforts? To answer these questions, we focus on building deep learning models with the necessary inductive biases and flexibility to learn nuanced features when given a large enough dataset, as opposed to hand-crafted rules that need revisiting to generalise to different time periods, writing styles, genres, or even languages. Moreover, we present a separate evaluation of pipeline's each component.

[*]Work done during internship at Amazon

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5820 - 5829
May 22-27, 2022 ©2022 Association for Computational Linguistics

2 Related Work

Attributing speakers to direct speech is a common problem for two related domains: news and literature. However, previous work (O'Keefe et al., 2012) has shown that models do not generalise to both domains. Their best model obtained an accuracy of 92.4% and 84.1% on their two newswire datasets, whilst they only found a 53.3% accuracy when evaluating the same model on a literature dataset. We therefore focus on summarising progress in quote attribution for literary fiction.

The early models targeting literary texts (Glass and Bangay, 2007) were based on the identification of speech verbs and their actors. However, the proportion of dialogues accompanied by a speech-verb and an explicit mention to a character can be as low as 20% of the total quotes for some books. For this reason, consequent work (O'Keefe et al., 2012; Elson and McKeown, 2010) shifted their focus to attributing speakers to dialogues where the character is not explicitly mentioned, incorporating rules to exploit the sequential nature of conversations. These models could not improve the results of a simple Nearest Mention (NM) baseline that obtained a 53.3% accuracy on their test set.

Finally, current state-of-the-art models (Muzny et al., 2017; He et al., 2013) demonstrated how the simple nearest mention baseline could be beaten through a combination of rules and learning. Both models present analysis on a limited setup: i) they report performance on a test set comprised of two books – Jane Austen's *Emma*, and Anton Chekhov's *The Steppe*, – and have therefore not been bench-marked on a wider range of styles or time periods, and ii) they assume the ideal circumstance of a pre-compiled list of characters, with character aliases and genders provided. We relax the second assumption when we evaluate our model, to estimate the end-to-end performance on a more diverse dataset of fifteen books.

The task of speaker attribution is also closely related to other dialogue sequence problems. One such umbrella technique to solve these problems is *Dialogue State Tracking* or simply DST, where a system is tasked with estimating some conversation state variables usually the user's goals and intents. We are first to apply DST for the purpose of speaker attribution. Our proposed DST-based formulation requires modification to the utterance encoder with focus on non-dialogue context, and state-variable that can generalise to states not seen in the training set. We adapt a BERT-based DST model (Lai et al., 2020) to track the speaker for every single utterance instead of tracking the user's goals and intents.

Our work follows a similar line of thought to Ren et al. (2018); Lai et al. (2020), where the model is given a list of candidate intents (speakers in our case) embedded as inputs to the problem so that the model can generalise to unseen goals (speakers) during test time. The task of our model is to generate a score for each utterance against every candidate speaker.

To recap, we present an end-to-end pipeline for speaker to dialogue attribution that leverages recent advances in large pretrained Language Models casting the problem as a Dialogue State Tracking. We empirically show that our model is capable of generalising to different styles more reliably as compared to prior hand-crafted features-based systems. Further, we present this comparative study on literary texts ranging from the 1900s to the 2010s, which are more varied and diverse compared to past studies. Note that usually such dataset are effort-intensive to create and not publicly available due to lack of rights to redistribution, which makes the reported result very interesting for the wider community.

3 Dataset

Our annotations consist of two independent layers, one focusing on direct speech, and one focusing on clustering mentions that refer to the same character entity.

Example 1: Excerpt from 2001: A Space Odyssey. Annotated direct speech is in bold, and the annotated attributed character entity inside a blue box.

Poole was asleep, and *Bowman* was reading on the control deck, when *Hal* announced:

"Er—*Dave*, I have a report for you." HAL

"What's up?" DAVID BOWMAN

"We have another bad AE-35 unit. My fault predictor indicates failure within twenty-four hours." HAL

For the first layer, the annotator selects the span of text representing a character's direct speech. It is usually found within quotation marks, but this is not a necessary or sufficient requirement. The annotator then attributes a character entity to the utterance. Example 1 presents a typical conversation with instances of coreference (Dave and Bowman both refer to David Bowman), and implicit attribution (third and fourth paragraphs) where no character is explicitly mentioned by the narrator.

The second layer of annotations focuses on characters and their mentions. We follow Bamman et al. (2014) and distinguish character *mentions* (e.g. Dave, David, Bowman, Dr. Bowman) in the text and character *entities* (e.g. DAVID BOWMAN), to which mentions refer to. See the text in italics within Example 1 to find some of the annotated character mentions. Note that we don't include pronominal mentions. These mentions are then clustered per book into character entities by the annotators.

We annotate a collection of 15 books sampled from the most popular titles from time epochs 1881 - 2018. The annotation is carried out by 3 expert English native annotators, each reading the book in sequential order, over a BRAT [1] based annotation interface. In case the annotation from any single annotator is different, a master annotator goes through the cases and makes the correction, resulting in a dataset with a very high agreement (Cohen's Kappa greater than 0.9). Across the books, the number of annotated dialogues varies from 200 to 5000 and characters from less than 10 to 200. We would refer these books by IDs 1 through 15 and the existing 3 books as E1 (*Emma*), E2 (*Pride and Prejudice*) and E3 (*The Steppe*).

4 Model components

We divide the model into three main tasks: identifying quotes, extracting unique characters and their aliases, and attributing dialogues to the extracted characters. Our goal is to reduce the amount of hand-crafted rules (usually heavily biased to the small subset of documents of prior studies) where performance can be improved, and allow the model to learn nuanced features that allow it to generalise better when given a large enough dataset.

We first introduce our direct speech identification component, which is purely rule-based due to the simplicity of the problem and since improving this aspect is not part of our core contribution. Afterward, we focus on identifying characters and compare NER and coreference resolution deep learning models to simple rule-based systems. Finally, we discuss the focal aspect of our contribution - a DST architectural adaptation to solve the speaker attribution of quotes.

4.1 Direct Speech identification

Direct speech in fiction is usually denoted with quotation marks, although there are exceptions such as Ali Smith's *Summer*, where speech marks are completely removed and dialogues blend in with the rest of the text, or Joyce's *Ulysses* that introduces speech with dashes. Here we ignore such instances, which are not present in our dataset, and focus on the most common case where direct speech is marked by quotation marks. Further, we find that for English over ≈ 95 % of the dialogues (as analysed over a large collection of popular books) follow open-close quotation-pair variation. See Steinbach et al. (2011) for an in-depth review of the topic.

In the case of extracting quotation marks, simple rules can achieve almost perfect performance. As in O'Keefe et al. (2012), we use a regular expression to detect opening and closing quotation marks that denote the presence of direct speech.

4.2 Character identification

Although characters are central to most literary analyses, identifying them automatically from a novel remains an unsolved problem. We split the character identification task into: i) identifying mentions in the text that refer to characters, and ii) clustering those mentions into unique character entities. Similar to direct speech identification, we do not focus on improving the architecture for character identification. As both of these form input to our core DST module, we re-purpose the best of existing techniques. However, unlike previous studies, we do analyse and report the impact of these components on end-to-end performance to guide future research.

Extracting entities from text is normally done over short documents, such as Wikipedia pages. But literature brings unique challenges to the field: novels tend to be long documents, demanding efficient algorithms, and requiring models to be able to link far apart mentions.

We present an evaluation of Named Entity Recognition (NER) to detect mentions, together with the effectiveness of coreference resolution to cluster character mentions into entities. We find that although NER achieves a similar performance to a simple rule-based system, coreference resolution's performance on clustering characters is significantly poorer than a simple rule-based character clustering technique, and future work should fo-

[1] https://brat.nlplab.org

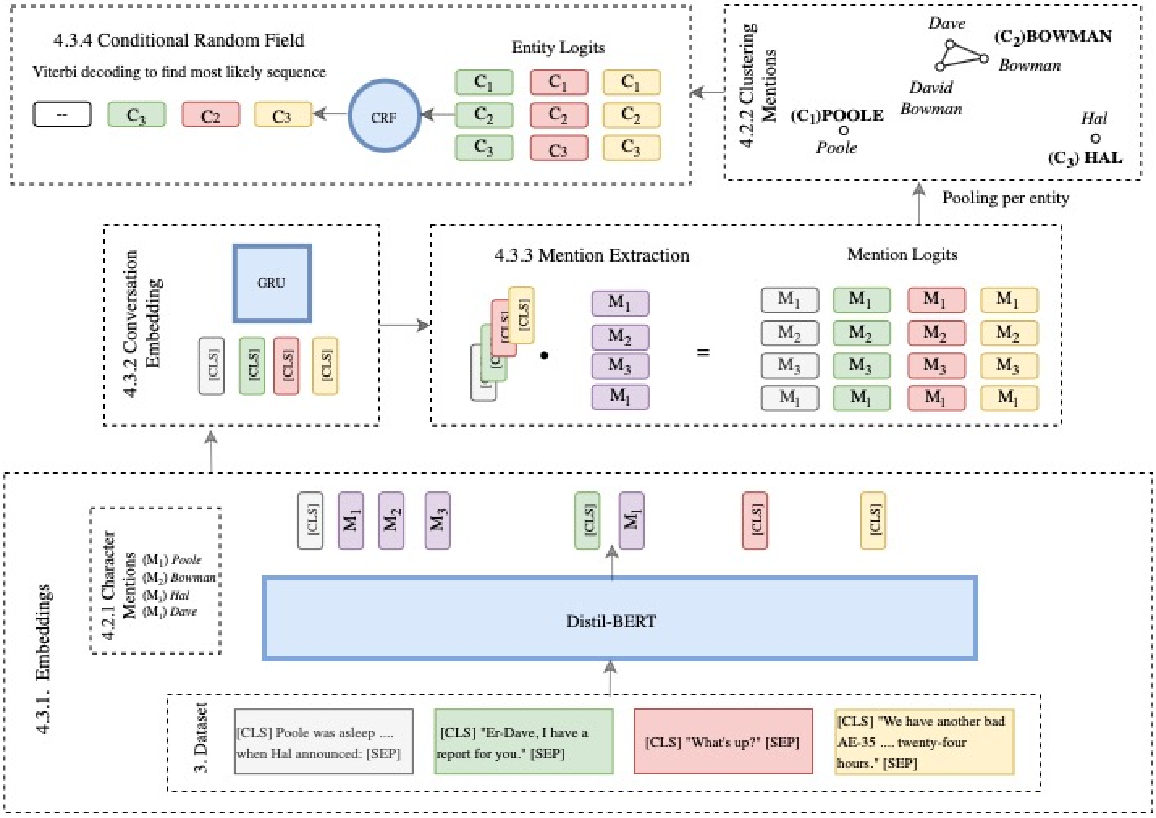

Figure 1: Diagram of our speaker attribution pipeline. Utterances in a conversation and mentions to characters are first embedded by using Distil-BERT. The extracted utterances are processed by a Gated Recurrent Network, and later combined with the embedded mentions through a dot product that results in the Mention Logits. Finally, we use the information on how mentions cluster into character entities to pool the maximum values of the Mention Logits by entity. The result is denoted by Entity Logits, and it is sent to a Conditional Random Field that generates a prediction by applying the Viterbi decoding algorithm.

cus on addressing this problem through techniques developed explicitly for the literature domain.

4.2.1 Identifying character mentions

We present a comparison between an out-of-domain NER model, trained on the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003) with a simple rule-based baseline that focuses on identifying all the characters that speak explicitly. It finds the subject of the narrator's explicit dialogue attribution signals (defined by the 40 most frequent speech denoting verbs such as *said, answered, ...*).

4.2.2 Clustering character mentions into entities

In Table 1, we show a summary of the aliases variations found in our dataset and their frequency. Since our core contribution is not to improve character clustering, and our dataset of character aliases is small, we do not develop a custom model for it and merely compare two different clustering techniques: i) an out-of-domain coreference resolution system, and ii) a simple set of rules that cluster characters according to their names. In both cases, we build a graph where nodes are character mentions, and edges are attached to all two compatible nodes. See the top right box on Clustering Mentions in Figure 1 for an example of such graph.

Type	Freq	Example ([mentions] → entity)
Full Name Variation	67%	[Harry, Potter, Mr. Potter, Harry Potter] → HARRY POTTER)
Dimunitives	7%	[Lizzy, Eliza] → ELIZABETH
Professional	15.5%	[the cook] → JOHN KING SILVER
Relational	5.5%	[her father] → MR. BENNET
Others	5%	[the cimmerian] → CONAN

Table 1: Summary of Character Name Variations for 50 randomly sampled characters over our dataset.

On one hand, in the case of coreference resolution, two nodes are compatible if they appear together at least in two coreferent clusters. Clusters of characters are formed by finding all disconnected subgraphs within the graph.

On the other hand, although character aliases can be of any kind and might not be related to each other by name, most of the ones appearing in literature are variations of the character's full name (See Table 1). We build a rule-based algorithm that deems two names incompatible if,
i) The first names of the characters are different (the first names do not match exactly) or the shorter first name is not exactly inside the longest one. This also takes care of a few diminutive forms (like Eliza for Elizabeth).
ii) Both names contain a title which is different.
iii) Both names contain a surname which is different.

We split the graph into disconnected subgraphs of compatible names, and find those nodes that are ambiguous, i.e., nodes whose first neighbour connections contain more than one title, first name, or surname. Removing those nodes we can form unambiguous clusters of characters that share the same title, name, and surname.

As opposed to previous work (Bamman et al., 2014; Elson et al., 2010), where ambiguous names would be merged to the closest entity mentioned in the text, we allow ambiguous nodes to either form their own cluster, or be part of any of their first neighbour nodes' clusters. We use the text to resolve this ambiguity by finding the most mentioned cluster among the possible clusters in the 20 paragraph vicinity of the ambiguous mention. In this way, we can retain characters such as *Mrs. Bennet* in *Pride and Prejudice*, without merging them to other members of the Bennet family, since they are prominent enough to be given their own cluster.

4.3 Speaker attribution

In this section, we present an adaptation of Dialogue State Tracking to speaker attribution. In DST, it is a challenge to produce models that can work with dynamic ontologies and unseen slot values such that the user can request information on any slot (movies, restaurants, ...) and use any value (the type of food, the price, ...) that has not necessarily been seen at test time. In the same way, we can't simply use a general fixed tag set of characters beyond the level of an individual novel, since we want our model to generalise to unseen novels and unseen characters during test time. We will therefore embed the character's mentions within the inputs of the model as done in state-of-the-art DST (Lai et al., 2020). Below, we discuss in detail how we adapt DST to model speaker attribution in novels.

4.3.1 Inputs definitions

Although our dataset is annotated at the level of word spans, the odds are high that disconnected spans in the same paragraph are attributed to the same speaker. We find that this rule is violated on less than 5% of the paragraphs that contain more than one disconnected span. Therefore, as in He et al. (2013), our model will be trained on attributing speakers at the level of paragraphs.

Regarding the model inputs, we split the text into conversations. Denoting paragraphs with no direct speech as narratives, we segment conversations by restricting the number of intervening narratives between direct speech utterances to one. If more than one narrative separates two direct speech utterances, the conversation is split into two different conversations.

Given a set of n utterances that define a conversation, $\mathbf{u} = \{u_0, u_1, ..., u_{n-1}\}$, a set of l mentions to candidate characters, $\mathbf{m} = \{m_0, m_1, ..., m_{l-1}\}$, and a set of k candidate characters entities linked to those mentions, $\mathbf{c} = \{c_0, c_1, ..., c_{k-1}\}$, where $k <= l$, we wish to model the probability for each candidate character entity, c_i, being the speaker of a given utterance, u_j. This probability will be denoted as $P\left(c_i|u_j, \mathbf{u}\right)$.

Let's denote as ϕ the embedding model that transforms word tokens in vectors (equivalently $\mathbb{R}^{D_\epsilon}$ space), where D_ϵ is the output dimension of the embedding model. In this work, we chose a Distil-BERT model (Sanh et al., 2019) for ϕ.

As in Figure 1, we generate an embedding for every utterance in the conversation,

$$\boldsymbol{\epsilon}_{u_i} = \phi_{\text{Distil-BERT}}(u_i)_{[CLS]}, \qquad (1)$$

by selecting the embedding of the [CLS] token, whereas to encode the character's mentions we take the mean of the embedding of the tokens inside the mention. For a mention $m_j = \{m_j^0, m_j^1, ..., m_j^{t-1}\}$ of length t,

$$\boldsymbol{\epsilon}_{m_j} = \frac{1}{t}\sum_{T=0}^{t-1} \phi_{\text{Distil-BERT}}(u_i)_{m_j^T}. \qquad (2)$$

We denote the collection of all mentions embeddings by $\boldsymbol{M}$, a matrix of dimensions $D_\epsilon \times D_l$, where D_l is the number of mentions to candidate characters inside the conversation.

In the next section, we explain the three components of the model that take these embeddings and produce the probability of a character speaking: i) the conversation embedding module, that takes an utterance as input and produces its context-aware representation, ii) a character extraction module, that given the contextual representation of an utterance and the embedding of the candidate characters mentions, generates the logits for each candidate character entity and utterance, and iii) a sequential decoder component that learns the conversational turn patterns. Below, we define the architecture of each component in detail.

4.3.2 Conversation Embedding

The conversation embedding module consists of a Gated Recurrent Unit (Cho et al., 2014; Gers et al., 1999), ϕ_{GRU}, that encodes the content of the conversation,

$$\mathbf{h}_{i+1} = \phi_{\text{GRU}}\left(\boldsymbol{\epsilon}_{u_{i+1}}, \mathbf{h}_i\right), \quad (3)$$

where $\mathbf{h}_i$ is the GRU's hidden state of dimension D_ϵ which is randomly initialised at the beginning of the sequence.

4.3.3 Character mention extraction

This module processes the GRU's hidden states to extract the candidate character's logits. We take the dot product of the utterance embedding, processed by a fully connected network, with the mention embedding matrix, $\boldsymbol{M}$, to obtain the logits,

$$\mathbf{l}_i = \phi_{\text{FCN}}\left(\mathbf{h}_i\right) \cdot \boldsymbol{M}, \quad (4)$$

where $\mathbf{l}_i$, has dimensions D_l. This for whole conversation results in $\boldsymbol{L}$ of dimension $D_n \times D_l$ and are denoted by Mention Logits in Figure 1.

We can now combine the logits of different mentions that belong to the same character entity by max-pooling over character entities to get the Entity Logits, $\boldsymbol{E}$ with dimensions $D_n \times D_k$.

4.3.4 Learning to take turns with Conditional Random Fields

Implicitly, the model defined above assumes that labels are independent of each other, and that therefore the likelihood of a sequence of labels in a conversation, $\mathbf{y}$, can be expressed as the product of utterance-wise likelihoods,

$$P(\mathbf{y}|\mathbf{u}) = \prod_m p(y_k|u_1, ..., u_m). \quad (5)$$

However, characters speaking in a conversation follow certain turn-taking patterns that are common through literature, such as a two-party conversation in which the dialogues move back and forth between two characters. We add a linear chain Conditional Random Field (CRF) (Lafferty et al., 2001) to our model, to maximise the likelihood of a sequence of characters and relax the label independence assumption by allowing a target to depend on its immediate predecessor.

A CRF models the sequential likelihood as a combination of element-wise prediction, and a pairwise interaction term that models the probability of transitioning from label y_i to label y_j. In our particular implementation, the element-wise predictions are the Entity Logits, $\boldsymbol{E}$, and the pairwise interaction will be learned parameters,

$$P(\mathbf{y}|\mathbf{u}) = \exp\left(\sum_{n=0}^{N} E_n(y_n) + \sum_{n=0}^{N-1} V_{y_n, y_{n+1}}\right)/Z, \quad (6)$$

where Z represents the normalization factor, and V is generally a $D_k \times D_k$ dimensional matrix of learned weights known as transition matrix.

In our use-case, there is no specific label ordering that can generalise to unseen novels, and we thus reduce the degrees of freedom of the $D_k \times D_k$ transition matrix to two: the value of the diagonal, and the value of the off-diagonal elements. The first one controls the probability of the same speaker to continue speaking, whereas the second one varies the probability of a change in speaker. Note that this implies that we do not need to constrain the number of speaking characters. At inference, we find the most likely sequence of characters using the Viterbi algorithm (Viterbi, 1967).

5 Evaluation of the model components

In this section, we present both the evaluation of each separate component and the final evaluation of the entire pipeline.

5.1 Direct Speech identification

To compare ground truth direct speech with our extracted quotes, we define a True Positive as an exact match between our selected text and the ground truth. With this definition, the F1-score achieved by our quote identification module is 0.98 ± 0.01, when evaluated against our entire dataset. We find that common errors are the identification of quoted text that has a different purpose other than direct

Model	F1-Score	Precision	Recall
NER	0.74 ± 0.1	0.82 ± 0.08	0.69 ± 0.14
Rule-based	0.78 ± 0.1	0.85 ± 0.07	0.76 ± 0.11

Table 2: Evaluation of the character mention component, compared to NER (Wolf et al., 2020). We show the average of all books and their standard deviation.

Model	F1-Score	Precision	Recall
Coreference	0.73 ± 0.1	0.97 ± 0.08	0.60 ± 0.14
Rule-based	0.86 ± 0.08	0.94 ± 0.07	0.79 ± 0.11

Table 3: Evaluation of the character clustering component, compared to coreference resolution (Wolf et al., 2020). We show the average of all books and their standard deviation.

speech, such as emphasising a word, naming a title, or marking written text, such as a letter, that no one is reading out loud.

5.2 Character identification

We evaluate separately the effects of identifying character mentions and clustering mentions into distinct entities. Since our model aims to resolve dialogue attribution and characters that are mentioned more often tend to also speak more often, we show precision, recall and F1-score weighted by the number of times a mention appears through the text. In this way, we make sure that we are identifying the main characters in the text at the cost of missing rarer ones. The evaluation is shown in Table 2. NER and our simple rule-based model show a similar performance, although overall the rule-based system works better.

Finally, we evaluate character clustering on oracle mentions using the B^3 measures of precision, recall and F1-score (Amigó et al., 2009). The results are reported in Table 3. The performance of coreference resolution is significantly worse than the simple naming rules we developed. By using name compatibility we achieve a high precision but a low recall, meaning that the clusters we create tend to contain elements of the same class, but are incomplete; a character might be split into several different clusters. This is because we only cluster characters from variations of their names, and therefore all other cases shown in Table 1 are considered as separate entities. As mentioned in Section 4.2.2, the coreference resolution model fails at linking two mentions to the same character that are far apart, and therefore produces a system with lower recall.

5.3 Speaker attribution

We train and evaluate the speaker attribution task on oracle direct speech, mentions and character clusters. We compute precision, recall and F1-Score, all weighted averages, of the character entities attributed to each span of direct speech.

Model	F1-Score	Precision	Recall
Our model	0.78 ± 0.06	0.81 ± 0.06	0.77 ± 0.06
NM	0.54 ± 0.08	0.57 ± 0.06	0.54 ± 0.08

Table 4: Evaluation of the speaker attribution component, compared to a baseline nearest mention (NM).

The resulting evaluation is shown in Table 4. We show a comparison with a baseline model that selects the nearest mention to either left or right of the quote. To include a thorough evaluation despite the small size of our dataset, we have trained the model in a leave-one-out fashion for all books for which we annotated more than $1,000$ paragraphs, together with the three publicly available books released by Muzny et al. (2017). In Table 4, we report average values and standard deviation over the 11 books.

Moreover, we show an ablation study in Table 5, computed on only one train, validation and test split. Next to the overall F1-Score we show the performance on the model by type of signal where a sample is: i) explicit, if the character is mentioned on the same paragraph as the quote, ii) implicit, if there is no narrator context accompanying the quote. Note that not all quotes fall in either of these categories.

5.4 End-to-end evaluation

Finally, the entire pipeline is evaluated as a clustering overlap problem through the B^3 clustering metric. A cluster is defined by the set of quotations attributed to the same character entity. If the quote has been incorrectly identified as a quote by the model, it forms part of a misidentified cluster. On the other hand, if we haven't identified one of the true quotes, we also label it as another kind of misidentification.

In Figure 2, we show a full pipeline comparison of our model to the state-of-the-art model presented in Muzny et al. (2017) [2]. Our model improves over Muzny et al. (2017) by an average of 50% in F1-

[2] We ran their publicly available code on our dataset, the code can be found here https://nlp.stanford.edu/~muzny/quoteli.html

	F1-Score	Accuracy Explicit	Accuracy Implicit
Distil-BERT (DB)	0.7 ± 0.06	0.89 ± 0.03	0.37 ± 0.03
DB + GRU	0.77 ± 0.07	0.94 ± 0.04	0.45 ± 0.05
DB + CRF	0.75 ± 0.09	0.94 ± 0.05	0.43 ± 0.07
DB + GRU + CRF	0.80 ± 0.06	0.95 ± 0.05	0.6 ± 0.08

Table 5: Summary of the ablation study results. We show the effect of removing different component of the models, where average values and standard deviations are computed over three test books: 14, 15, and E3 (*The Steppe*).

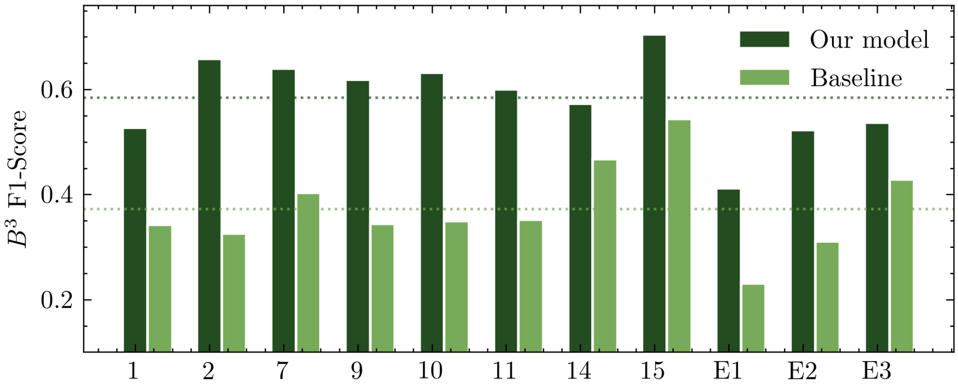

Figure 2: Chart showing F1-score values for different books training the model in a leave-one-out fashion. Average values for both our model and the baseline are shown in dotted lines.

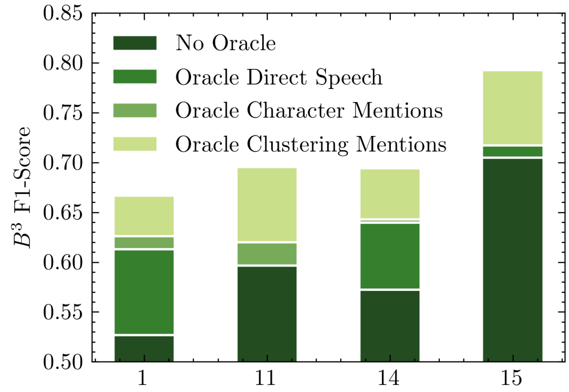

Figure 3: Stacked chart showing the F1-score when different components of the model are replaced by their oracle (ground truth) value.

score, and achieves a more consistent performance across different styles and time periods.

We also show the effect of replacing the different components with Oracle data for a subsample of the dataset in Figure 3. We can see that different components play a different role by book, whereas improving the mention extraction stage can be of crucial importance for some books (14 and 15 part of text split; and 1 and 11 part of train split), character clustering has a larger effect on others. However, the dominant effect is still the Speaker attribution model.

6 Conclusions and Future Work

We have presented a speaker attribution pipeline for novels that does not rely on pre-compiled lists of characters and that performs consistently across different writing styles and time periods. Our main contribution has been to develop the first deep learning model for speaker attribution, based on previous Dialogue State Tracking approaches. Our deep learning model has the flexibility to learn nuanced features from data, compared to previous work that relied on rules or hand-crafted features. Training our model on a small dataset composed of 15 different novels, we find that it outperforms the model presented in Muzny et al. (2017) by an average of 50% F1-score. In the future, we hope to improve our model by: training it on a larger and more varied dataset, and training the model on speaker attribution together with the related task of coreference resolution.

We have also presented an error analysis on the different components necessary to perform the end-to-end goal of attributing characters to their direct speech utterances in novels: i) direct speech identification, ii) character extraction, and iii) speaker attribution. We have shown the need to produce literature-domain specific models targeting character extraction in order to improve the accuracy of current systems.

References

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486.

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.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation.

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 K. Elson and Kathleen R. McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1013–1019. AAAI Press.

F. A. Gers, J. Schmidhuber, and F. Cummins. 1999. Learning to forget: continual prediction with LSTM. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2.

Kevin Glass and Shaun Bangay. 2007. A naive salience-based method for speaker identification in fiction books. In *In PRASA 2007: Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa*, pages 1–6.

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.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan L. Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1534–1544.

Prashant Jayannavar, Apoorv Agarwal, Melody Ju, and Owen Rambow. 2015. Validating literary theories using automatic social network extraction. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 32–41, Denver, Colorado, USA. Association for Computational Linguistics.

Eve Kraicer and Andrew Piper. 2019. Social characters: The hierarchy of gender in contemporary english-language fiction. *Journal of Cultural Analytics*, 1(1).

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Comput. Surv.*, 52(5).

Faisal Ladhak, Bryan Li, Yaser Al-Onaizan, and Kathleen R. McKeown. 2020. Exploring content selection in summarization of novel chapters. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5043–5054.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

T. M. Lai, Q. Hung Tran, T. Bui, and D. Kihara. 2020. A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8034–8038.

Tuan Manh Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2020. A simple but effective bert model for dialog state tracking on resource-limited systems. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 8034–8038. IEEE.

Grace Muzny, Michael Fang, Angel Chang, and Dan Jurafsky. 2017. 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.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. In *Proceedings of the 2018 Conference on Empirical Meth-*

ods in Natural Language Processing, pages 2780–2786, Brussels, Belgium. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Markus Steinbach, Jorg Meibauer, and Elke Brendel, editors. 2011. *Understanding Quotation*. Mouton Series in Pragmatics [MSP] ; 7. De Gruyter Mouton,, Berlin.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

William E Underwood, David Bamman, and Sabrina Lee. 2018. The transformation of gender in english-language fiction. *Journal of Cultural Analytics*, 1(1).

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Appendix

Model hyper-parameters

In this appendix we describe the training procedure and hyper-parameters for the DST model. During training, we use early stopping with a patience of 6 epochs, a batch size of 3 conversations, and cross-entropy as a loss function. The initial learning rate is set to $1e{-}5$ for the BERT model layers, and $1.e{-}4$ for all the others. We use a linear schedule with warmup, with the number of warmup steps set to 0. Moreover, we limit the maximum conversation length to 45 utterances. Regarding the model's architecture, the recurrent network is a bidirectional GRU with only one layer of dimension 768, and the fully connected network contains also a single layer with the same dimensions. A dropout of 0.2 is applied to both the output of BERT and the output of the fully connected layer. The model was trained on a single NVIDIA Tesla V100 SXM2 16 GB GPU.