Leveraging Discourse Information Effectively for Authorship Attribution*

Elisa Ferracane[1], **Su Wang**[1,2] and **Raymond J. Mooney**[3]

[1]Department of Linguistics, The University of Texas at Austin
[2]Department of Statistics and Data Science, The University of Texas at Austin
[3]Department of Computer Science, The University of Texas at Austin
elisa@ferracane.com, shrekwang@utexas.edu, mooney@cs.utexas.edu

Abstract

We explore techniques to maximize the effectiveness of discourse information in the task of authorship attribution. We present a novel method to embed discourse features in a Convolutional Neural Network text classifier, which achieves a state-of-the-art result by a significant margin. We empirically investigate several featurization methods to understand the conditions under which discourse features contribute non-trivial performance gains, and analyze discourse embeddings.[1]

1 Introduction

Authorship attribution (AA) is the task of identifying the author of a text, given a set of author-labeled training texts. This task typically makes use of stylometric cues at the surface lexical and syntactic level (Stamatatos et al., 2015), although Feng and Hirst (2014) and Feng (2015) go beyond the sentence level, showing that discourse information can help. However, they achieve limited performance gains and lack an in-depth analysis of discourse featurization techniques. More recently, convolutional neural networks (CNNs) have demonstrated considerable success on AA relying only on character-level n-grams (Ruder et al., 2016; Shrestha et al., 2017). The strength of these models is evidenced by findings that traditional stylometric features such as word n-grams and POS-tags do not improve, and can sometimes even hurt performance (Ruder et al., 2016; Sari et al., 2017). However, none of these CNN models make use of discourse.

Our work builds upon these prior studies by exploring an effective method to (i) featurize the discourse information, and (ii) integrate discourse features into the best text classifier (i.e., CNN-based models), in the expectation of achieving state-of-the-art results in AA.

Feng and Hirst (2014) (henceforth F&H14) made the first comprehensive attempt at using discourse information for AA. They employ an entity-grid model, an approach introduced by Barzilay and Lapata (2008) for the task of ordering sentences. This model tracks how the grammatical relations of salient entities (e.g., `subj`, `obj`, etc.) change between pairs of sentences in a document, thus capturing a form of discourse coherence. The grid is summarized into a vector of transition probabilities. However, because the model only records the transition between two consecutive sentences at a time, the coherence is *local*. Feng (2015) (henceforth F15) further extends the entity-grid model by replacing grammatical relations with discourse relations from Rhetorical Structure Theory (Mann and Thompson, 1988, RST). Their study uses a linear-kernel SVM to perform pairwise author classifications, where a non-discourse model captures lexical and syntactic features. They find that adding the entity-grid with grammatical relations enhances the non-discourse model by almost 1% in accuracy, and using RST relations provides an improvement of 3%. The study, however, works with only one small dataset and their models produce overall unremarkable performance (∼85%). Ji and Smith (2017) propose an advanced Recursive Neural Network (RecNN) architecture to work with RST in the more general area of text categorization and present impressive results. However, we suspect that the massive number of parameters of RecNNs would likely cause overfitting when working with smaller datasets, as is often the case in AA tasks.

*The first two authors contributed equally to this work.
[1]https://github.com/elisaF/authorship-attribution-discourse

(1) [My father]$_S$ was a clergyman of the north of England, [who]$_O$ was deservedly respected by all who knew [him]$_O$; and, in his younger days, lived pretty comfortably on the joint income of a small incumbency and a snug little property of his own.
(2) [My mother]$_S$, who married [him]$_O$ against the wishes of her friends, was a squire's daughter, and a woman of spirit.
(3) In vain it was represented to [her]$_X$, that if [she]$_S$ became [the poor parson's]$_X$ wife, [she]$_S$ must relinquish her carriage and her lady's-maid, and all the luxuries and elegancies of affluence; which to [her]$_X$ were little less than the necessaries of life.

Table 1: Excerpt of 19^{th}-century novel where sentences are labeled with the salient entities and their grammatical relations (subject **s**, object **o**, other relation **x**). A salient entity is a noun phrase coreferred to at least two times in a document.

	SS	SO	SX	S-	OS	OO	OX	O-	XS	XO	XX	X-	-S	-O	-X	–
d_1	0.25	0.25	0	0	0	0	0.25	0	0	0	0	0	0.25	0	0	0

Table 2: The probability vector for the excerpt in Table 1 capturing transition probabilities of length 2.

In our paper, we opt for a state-of-the-art character-bigram CNN classifier (Shrestha et al., 2017). We choose to use the entity-grid model because we find it helps avoid overfitting[2] (adding typical stylometric features such as word n-grams and POS tags results in overfitting) and further captures coreference chains, which we show are critical to improving performance on this task (see Section 5). We investigate various ways in which the discourse information can be featurized and integrated into the CNN. Specifically,

- *Featurization*. We attempt to capture a more *global* discourse coherence by modeling the entire sequence of relations in a document for every salient entity, instead of only the relations between pairs of sentences.

- *Feature integration*. Using a neural network architecture allows us to explore embedding the relations from the entity-grid model,[3] rather than only exploiting a vector of relation probabilities.

We explore these questions using two approaches to represent salient entities: grammatical relations, and RST discourse relations. We apply these models to datasets of varying sizes and genres, and find that adding any discourse information improves AA consistently on longer documents, but has mixed results on shorter documents. Further, embedding the discourse features in a parallel CNN at the input end yields better performance than concatenating them to the output layer as a feature vector (Section 3). The global featurization is more effective than the local one. We also show that SVMs, which can only use discourse probability vectors, neither produce a competitive performance (even with fine-tuning), nor generalize in using the discourse information effectively.

2 Background

Entity-grid model. Typical lexical features for AA are relatively superficial and restricted to within the same sentence. F&H14 hypothesize that discourse features beyond the sentence level also help authorship attribution. In particular, they propose an author has a particular style for representing entities across a discourse. Their work is based on the entity-grid model of Barzilay and Lapata (2008) (henceforth B&L).

The entity-grid model tracks the grammatical relation (`subj`, `obj`, etc.) that salient entities take on throughout a document as a way to capture local coherence . A salient entity is defined as a noun phrase that co-occurs at least twice in a document. Extensive literature has shown that subject and object relations are a strong signal for salience and it follows from Centering Theory that you want to avoid rough shifts in the center (Grosz et al., 1995; Strube and Hahn, 1999). B&L thus focus on whether a salient entity is a subject (**s**), object (**o**), other (**x**), or is not present (**-**) in a given sentence, as illustrated in Table 1. Every sentence in a document is encoded with the grammatical relation of all the salient entities, resulting in a grid

[2]Primarily compared to previous work where discourse trees are modeled with Recursive Neural Nets (Ji and Smith, 2017).

[3]Tien Nguyen and Joty (2017) are the first to propose applying embeddings in modeling local coherence (for the coherence judgment task). Our methods roughly subsume theirs, which correspond to our GR CNN2-DE (global) model (Section 3). This scheme did not come out on top in our AA tasks.

similar to Table 3.

	father	mother
(1)	s	-
(2)	o	s
(3)	x	s

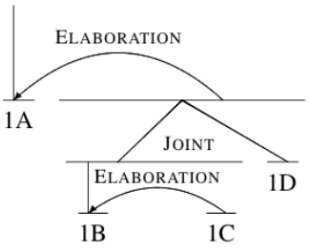
Table 3: The entity grid for the excerpt in Table 1, where columns are salient entities and rows are sentences. Each cell contains the grammatical relation of the given entity for the given sentence (subject **s**, object **o**, another grammatical relation **x**, or not present **-**). If an entity occurs multiple times in a sentence, only the highest-ranking relation is recorded.

The local coherence of a document is then defined on the basis of local entity transitions. A local entity transition is the sequence of grammatical relations that an entity can assume across n consecutive sentences, resulting in $\{s,o,x,-\}^n$ possible transitions. Following B&L, F&H14 consider sequences of length n=2, that is, transitions between two consecutive sentences, resulting in 4^2=16 possible transitions. The probability for each transition is then calculated as the frequency of the transition divided by the total number of transitions. This step results in a single probability vector for every document, as illustrated in Table 2.

B&L apply this model to a sentence ordering task, where the more coherent option, as evidenced by its transition probabilities, was chosen. In authorship attribution, texts are however assumed to already be coherent. F&H14 instead hypothesize that an author unconsciously employs the same methods for describing entities as the discourse unfolds, resulting in discernible transition probability patterns across multiple of their texts. Indeed, F&H14 find that adding the B&L vectors increases the accuracy of AA by almost 1% over a baseline lexico-syntactic model.

RST discourse relations. F15 extends the notion of tracking salient entities to RST. Instead of using grammatical relations in the grid, RST discourse relations are specified. An RST discourse relation defines the relationship between two or more elementary discourse units (EDUs), which are spans of text that typically correspond to syntactic clauses. In a relation, an EDU can function as a nucleus (e.g., `result.N`) or as a satellite (e.g., `summary.S`). All the relations in a document then form a tree as in Figure 1.[4]

[4]For reasons of space, only the first sentence of the excerpt is illustrated.



[My father was a clergyman of the north of England,]^{1A} [who was deservedly respected by all]^{1B} [who knew him;]^{1C} [and, in his younger days, lived pretty comfortably on a joint of a small incumbency and a snug little property of his own.]^{1D}

Figure 1: RST tree for the first sentence of the excerpt in Table 1.

F15 finds that RST relations are more effective for AA than grammatical relations. In our paper, we populate the entity-grid in the same way as F15's "Shallow RST-style" encoding, but use fine-grained instead of coarse-grained RST relations, and do not distinguish between intra-sentential and multi-sentential RST relations, or salient and non-salient entities. We explore various featurization techniques using the coding scheme.

CNN model. Shrestha et al. (2017) propose a convolutional neural network formulation for AA tasks (detailed in Section 3). They report state-of-the-art performance on a corpus of Twitter data (Schwartz et al., 2013), and compare their models with alternative architectures proposed in the literature: (i) SCH: an SVM that also uses character n-grams, among other stylometric features (Schwartz et al., 2013); (ii) LSTM-2: an LSTM trained on bigrams (Tai et al., 2015); (iii) CHAR: a *Logistic Regression* model that takes character n-grams (Stamatatos, 2009); (iv) CNN-W: a CNN trained on word embeddings (Kalchbrenner et al., 2014). The authors show that the model CNN2[5] produces the best performance overall. Ruder et al. (2016) apply character n-gram CNNs to a wide range of datasets, providing strong empirical evidence that the architecture generalizes well. Further, they find that including word n-grams in addition to character n-grams reduces performance, which is in agreement with Sari et al. (2017)'s findings.

[5]Shrestha et al. (2017) test two variants of CNN models: **CNN1/CNN2** for unigram/bigram character CNNs respectively.

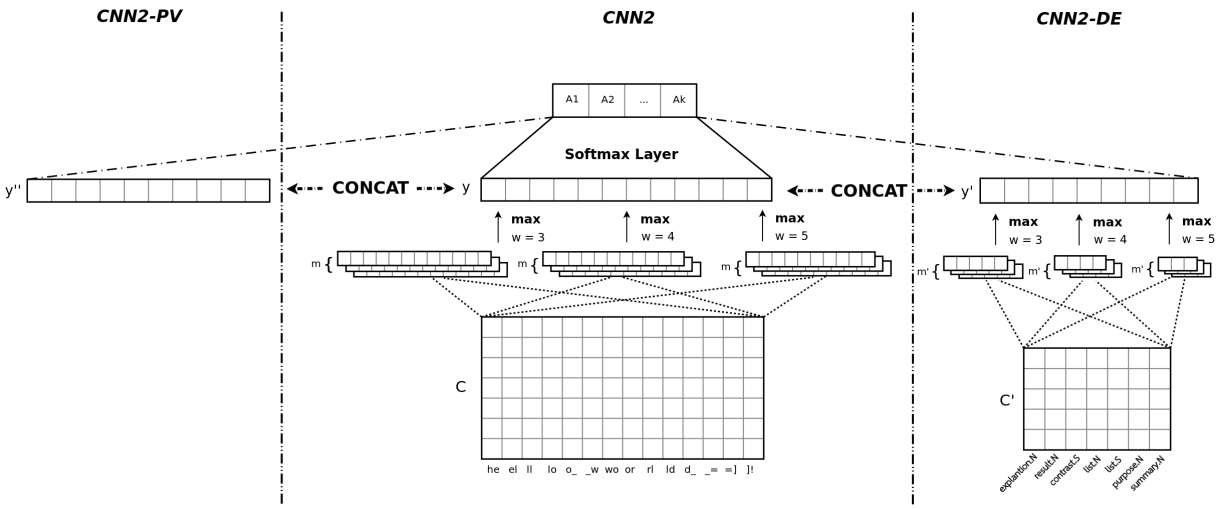


Figure 2: The bigram character CNN models

3 Models

Building on Shrestha et al. (2017)'s work, we employ their character-bigram CNN (**CNN2**)[6], and propose two extensions which utilize discourse information: (i) CNN2 enhanced with relation *probability vectors* (**CNN2-PV**), and (ii) CNN2 enhanced with *discourse embeddings* (**CNN2-DE**). The CNN2-PV allows us to conduct a comparison with F&H14 and F15, which also use relation probability vectors.

CNN2. CNN2 is the baseline model with no discourse features. Illustrated in Figure 2 (center), it consists of (i) an embedding layer, (ii) a convolution layer, (iii) a max-pooling layer, and (iv) a softmax layer. We briefly sketch the processing procedure and refer the reader to (Shrestha et al., 2017, Section 2) for mathematical details.

The network takes a sequence of character bigrams $\boldsymbol{x} = \langle x_1, \ldots, x_l \rangle$ as input, and outputs a multinomial ϕ over class labels as the prediction. The model first looks up the embedding matrix to produce a sequence of embeddings for $\boldsymbol{x}$ (i.e., the matrix C), then pushes the embedding sequence through convolutional filters of three bigram-window sizes $w = 3, 4, 5$, each yielding m feature maps. We then apply the *max-over-time* pooling (Collobert et al., 2011) to the feature maps from each filter, and concatenate the resulting vectors to obtain a single vector $\boldsymbol{y}$, which then goes through the softmax layer to produce predictions.

CNN2-PV. This model (Figure 2, left+center) featurizes discourse information into a probability vector (PV). The discourse features come in two flavors: (i) grammatical relations (GR), and (ii) RST discourse relations (RST)[7]. For both types of discourse features, an entity grid is first constructed to identify salient entities[8]. Recall each row in the grid is a sentence, and each column is a salient entity. The values of each cell in the grid are then populated differently, depending on which flavor of discourse feature is used.

For GR features, the entity grid is populated with the grammatical relation of each entity in each sentence. The entity grid is then collapsed into a single probability vector as shown in Table 2. The GR feature vector thus consists of a sequence of *grammatical relation transitions* derived from the entity grid, e.g., $\langle \texttt{sx}, \texttt{xs}, \texttt{ss}, \ldots \rangle$. The vector is a distribution over all the grammatical role transitions, i.e., $\langle p(\texttt{sx}), p(\texttt{xs}), p(\texttt{ss}), \ldots \rangle$.

For RST features, the entity grid is populated with the RST relation and nulcearity of the entity, and additionally the relations and nuclearity of the main EDUs in the current and previous sentence (as in Feng (2015)). We do not encode the entire RST tree since prior work has shown better performance with underspecified trees (Ji and Smith, 2017; Hogenboom et al., 2015). The RST features are represented as *RST discourse relations* with their nuclearity, e.g., $\langle \texttt{definition.N}, \texttt{attribution.S}, \ldots \rangle$. The probability vector is a distribution

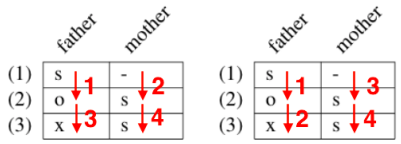
[6] Our preliminary experiments found that using character n-gram orders higher than 2 performed worse, likely due to the increased number of features and overfitting.

[7] Using RST Parser from Ji and Eisenstein (2014).

[8] Using neural coreference resolver, dependency parser in Stanford Core NLP (Clark and Manning, 2016).

Dataset	# authors	mean words/auth	range words/auth
NOVEL-9	9	376,242	124K-1M
NOVEL-50	50	709,880	184K-2.1M
IMDB62	62	349,004	9.8K-75K

Table 4: Statistics for datasets.



(a) local (b) global

Figure 3: Two variants for creating sequences of grammatical relation transitions in an entity grid.

over all the RST discourse relations, i.e., $\langle p(\texttt{definition.N}), p(\texttt{attribution.S}), \ldots \rangle$

Denoting the discourse feature vector with $\boldsymbol{y}''$, we construct the pooling vector $\boldsymbol{y}$ for the char-bigrams, and concatenate $\boldsymbol{y}''$ to $\boldsymbol{y}$ before feeding the resulting vector to the softmax layer.

CNN2-DE. In this model (Figure 2, center+right), we embed discourse features in high-dimensional space (similar to char-bigram embeddings). Let $\boldsymbol{z} = \langle z_1, \ldots, z_{l'} \rangle$ be a sequence of discourse features[9], we treat it in a similar fashion to the char-bigram sequence $\boldsymbol{x}$, i.e. feeding it through a "parallel" convolutional net (Figure 2 right). We set the embedding size to the average number of relations, then either pad or truncate. The operation results in a pooling vector $\boldsymbol{y}'$. We concatenate $\boldsymbol{y}'$ to the pooling vector $\boldsymbol{y}$ (which is constructed from $\boldsymbol{x}$) then feed $[\boldsymbol{y}; \boldsymbol{y}']$ to the softmax layer for the final prediction.

4 Experiments and Results

We begin by introducing the datasets (Section 4.1), followed by detailing the featurization methods (Section 4.2), the experiments (Section 4.3), and finally reporting results (Section 4.4).

4.1 Datasets

The statistics for the three datasets used in the experiments are summarized in Table 4.

novel-9. This dataset was compiled by F&H14: a collection of 19 novels by 9 nineteenth century British and American authors in the Project Gutenberg. To compare to F&H14, we apply the same resampling method (F&H14, Section 4.2) to correct the imbalance in authors by oversampling the texts of less-represented authors.

novel-50. This dataset extends novel-9, compiling the works of 50 randomly selected authors of the same period. For each author, we randomly select 5 novels for a total 250 novels.

IMDB62. IMDB62 consists of 62K movie reviews from 62 users (1,000 each) from the Internet Movie dataset, compiled by Seroussi et al. (2011). Unlike the novel datasets, the reviews are considerably shorter, with a mean of 349 words per text.

4.2 Featurization

As described in Section 2, in both the GR and RST variants, from each input entry we start by obtaining an entity grid.

CNN2-PV. We collect the probabilities of entity role transitions (in GR) or discourse relations (in RST) for the entries. Each entry corresponds to a probability distribution vector.

CNN2-DE. We employ two schemes for creating discourse feature sequences from an entity grid. While we always read the grid by column (by a salient entity), we vary whether we track the entity across a number of sentences (n rows at a time) or across the entire document (one entire column at a time), denoted as *local* and *global* readings respectively.

For the GR discourse features, in the case of local reading, we process the entity roles one sentence pair at a time (Figure 3, left). For example, in processing the pair (s_1, s_2), we find the first non-empty role r_{11} for entity $E1$ in s_1. If $E1$ also has a non-empty role r_{21} in the s_2, we collect the entity role transition $r_{11}r_{21}$. We then proceed to the following entity $E2$, until we process all the entities in the grid and move to the next sentence pair. For the global reading, we instead read the entity roles by traversing one column of the entire document at a time (Figure 3, right). The entity roles in all the sentences are read for one entity: we collect transitions for all the non-empty roles (e.g., $\texttt{so}$, but not $\texttt{s}-$).

For the RST discourse features, we process non-empty discourse relations also through either local or global reading. In the local reading, we read all the discourse relations in a sentence (a row) then

[9] The sequence comes in two variants, depending on the featurization technique, see Section 4.2.

move on to the next sentence.[10] In the global reading, we read in discourse relations for one entity at a time. This results in sequences of discourse relations for the input entries.

4.3 Experiments

Baseline-dataset experiments. All the baseline-dataset experiments are evaluated on novel-9. As a comparison to previous work (F15), we evaluate our models using a pairwise classification task with GR discourse features. In her model, each novel is partitioned into 1000-word chunks, and the model is evaluated with accuracy.[11] Surpassing F15's SVM model by a large margin, we then further evaluate the more difficult multi-class task, i.e., all-class prediction simultaneously, with both GR and RST discourse features and the more robust F1 evaluation. In this multi-class task, we implement two SVMs to extend F15's SVM models: (i) SVM2: a linear-kernel SVM which takes char-bigrams as input, as our CNNs, and (ii) SVM2-PV: an updated SVM2 which takes also probability vector features.

Further, we are interested in finding a performance threshold on the minimally-required input text length for discourse information to "kick in". To this end, we chunk each novel into different sizes: 200-2000 words, at 200-word intervals, and evaluate our CNNs in the multi-class condition.

Generalization-dataset experiments. To confirm that our models generalize, we pick the best models from the baseline-dataset experiments and evaluate on the novel-50 and IMDB62 datasets. For novel-50, the chunking size applied is 2000-word as per the baseline-dataset experiment results, and for IMDB62, texts are not chunked (i.e., we feed the models with the original reviews directly). For model comparison, we also run the SVMs (i.e., SVM2 and SVM2-PV) used in the baseline-dataset experiment. All the experiments conducted here are multi-class classification with macro-averaged F1 evaluation.

Model configurations. Following F15, we perform 5-fold cross-validation. The embedding sizes are tuned on novel-9 (multi-class condition): 50 for char-bigrams; 20 for discourse features. The learning rate is 0.001 using the Adam Optimizer (Kingma and Ba, 2014). For all models, we apply dropout regularization of 0.75 (Srivastava et al., 2014), and run 50 epochs (batch size 32). The SVMs in the baseline-dataset experiments use default settings, following F15. For the SVMs in the generalization-dataset experiments, we tuned the hyperparameters on novel-9 with a grid search, and found the optimal setting as: stopping condition `tol` is 1e-5, at a max-iteration of 1,500.

[10] We do not check the next sentences as in GR, because the discourse relations in one cell of the entity grid typically already capture relations beyond the sentence level.

[11] Averaged over all the author-author pair experiments.

MODEL	AVG.ACCURACY
Baseline	49.8
SVM (LexSyn)	85.5
SVM (LexSyn-PV)	86.4
CNN2	99.5
CNN2-PV	**99.8**

Table 5: Accuracy for pairwise author classification on the novel-9 dataset, using either a dumb baseline, an SVM with and without discourse to replicate F15, or a bigram-character CNN (CNN2) with and without discourse.

DISC.TYPE	MODEL	F1
None	SVM2	84.9
	CNN2	95.9
GR	SVM2-PV	85.7
	CNN2-PV	96.1
	CNN2-DE (local)	97.0
	CNN2-DE (global)	96.9
RST	SVM2-PV	85.9
	CNN2-PV	96.3
	CNN2-DE (local)	97.4
	CNN2-DE (global)	**98.5**

Table 6: Macro-averaged F1 score for multi-class author classification on the novel-9 dataset, using either no discourse (None), grammatical relations (GR), or RST relations (RST). These experiments additionally include the Discourse Embedding (DE) models for GR and RST.

4.4 Results

Baseline-dataset experiments. The results of the baseline-dataset experiments are reported in Table 5, 6 and Figure 4. In Table 5, Baseline denotes the dumb baseline model which always predicts the more-represented author of the pair. Both SVMs are from F15, and we report her results. SVM (LexSyn) takes character and word bi/trigrams and POS tags. SVM (LexSyn-PV) additionally includes probability vectors, similar to our CNN2-PV. In this part of the experiment, while the CNNs clear a large margin over SVMs (all differences

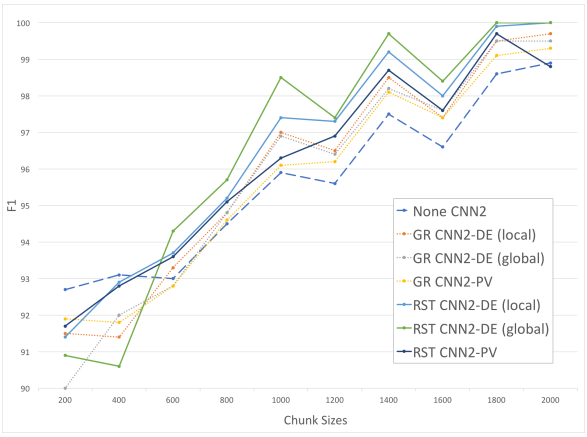


Figure 4: Macro-averaged F1 score for multi-class author classification on the novel-9 dataset in varied chunk sizes.

DISC. TYPE	MODEL	NOVEL-50	IMDB62
None	SVM2	92.9	90.4
	CNN2	95.3	91.5
GR	SVM2-PV	93.3	90.4
	CNN2-PV	95.1	90.5
	CNN2-DE (local)	96.9	90.8
	CNN2-DE (global)	97.5	90.9
RST	SVM2-PV	93.8	90.9
	CNN2-PV	95.5	90.7
	CNN2-DE (local)	97.7	91.4
	CNN2-DE (global)	**98.8**	**92.0**

Table 7: Macro-averaged F1 score for multi-class author classification on the large datasets, using either no discourse (None), grammatical relations (GR), or RST relations (RST).

are statistically significant at p<0.005), adding discourse in CNN2-PV brings only a small performance gain.

Table 6 reports the results from the multi-class classification task, the more difficult task. Here, probability vector features (i.e., PV) again fail to contribute much. The discourse embedding features, on the other hand, manage to increase the F1 score by a noticeable amount, with the maximal improvement seen in the CNN2-DE (global) model with RST features (by 2.6 points). In contrast, the discourse-enhanced SVM2-PVs increase F1 by about 1 point, with overall much lower scores in comparison to the CNNs. In general, RST features work better than GR features.

The results of the varying-sizes experiments are plotted in Figure 4. Again, we observe the overall pattern that discourse features improve the F1 score, and RST features procure superior performance. Crucially, however, we note there is no performance boost below the chunk size of 1000 for GR features, and below 600 for RST features. Where discourse features do help, the GR-based models achieve, on average, 1 extra point on F1, and the RST-based models around 2.

Generalization-dataset experiments. Table 7 summarizes the results of the generalization-dataset experiments. All reported statistical tests are t-test with a significance level of 0.05. First, we note that CNNs show a clear advantage over SVMs for all model variants on both datasets (confirmed significant for SVM2 vs. CNN2 with no discourse, SVM2-PV vs. CNN2-DE with GR and RST). On novel-50, most discourse-enhanced models significantly improve the performance of the baseline non-discourse CNN2 to varying degrees (significant for CNN2 with no discourse vs. CNN2-DE with GR and RST). The clear pattern again emerges that RST features work better, with the best F1 score evidenced in the CNN2-DE (global) model (3.5 improvement in F1) (significant for CNN2-DE with GR vs. CNN2-DE with RST). On IMDB62, as expected with short text inputs (mean=349 words/review), the discourse features in general do not add further contribution. Even the best model, CNN2-DE, brings only marginal improvement (not statistically significant), confirming our findings from varying the chunk size on novel-9, where discourse features did not help at this input size. However, the difference between the GR and RST variants for the IMDB CNN models are statistically significant. For the SVM models on both datasets, we note discourse features do not make noticeable improvements. On novel-50, SVM2-PV performs slightly better than the no-discourse SVM2 (by 0.4 with GR, 0.9 with RST features). On IMDB62, the same pattern persists with no gains for GR and 0.5 for RST features.

5 Analysis

General analysis. Overall, we have shown that discourse information can improve authorship attribution, but only when properly encoded. This result is critical in demonstrating the particular value of discourse information, because typical stylometric features such as word n-grams and POS tags do *not* add additional performance improvements (Ruder et al., 2016; Sari et al., 2017).

In addition, the type of discourse information and the way in which it is featurized are crucial to this performance improvement: RST features provide overall stronger improvement, and the global

TARGET EMBEDDING	TOP NEIGHBORS
explanation.N	interpretation.N, explanation.S, example.N, purpose.S, reason.N
background.N	circumstances.S, contrast.N, comparison.N, antithesis.S, elaboration.N
consequence.N	result.N, list.N, result.S, comment.N, summary.N

Table 8: Nearest neighbors of example embeddings with t-SNE clustering (top 5)

reading scheme for discourse embedding works better than the local one. The discourse embedding proves to be a superior featurization technique, as evidenced by the generally higher performance of CNN2-DE models over CNN2-PV models. With an SVM, where the option is not available, we are only able to use relation probability vectors to obtain a very modest performance improvement.

Further, we found an input-length threshold for utilizing discourse features is helpful (Section 4.4). Not surprisingly, discourse does not contribute on shorter texts. Many of the feature grids are empty for these shorter texts– either there are no coreference chains or they are not correctly resolved. Currently we only have empirical results on short novel chunks and movie reviews, but believe the finding would generalize to Twitter or blog posts.

Discourse embeddings. It does not come as a surprise that discourse-embedding-based models perform better than their relation-probability-based peers. The former (i) leverages the weight learning of the entire computational graph of the CNN rather than only the softmax layer, as the PV models do, and (ii) provides a more fine-grained featurization of the discourse information. Rather than merely taking a probability over grammatical relation transitions (in GR) or discourse relation types (in RST), in DE-based models we learn the dependency between grammatical relation transitions/discourse relations through the w-sized filter sweeps.

To further study the information encoded in the discourse embeddings, we performed `t-SNE` clustering (van der Maaten and Hinton, 2008) on them, using the best performing model CNN2-DE (global). We examined the closest neighbors of each embedding, and observed that similar discourse relations tend to go together (e.g., `explanation` and `interpretation`; `consequence` and `result`). Some examples are given in Table 8. However, it is unclear how this pattern helps improve classification performance. We intend to investigate this question in future work.

Global vs. Local featurization. As described in Section 4.2, the global reading processes all the discourse features for one entity at a time, while the local approach reads one sentence (or one sentence pair) at a time. In all the relevant experiments, global featurization showed a clear performance advantage (on average 1 point gain in F1). Recall that the creation of the grids (both GR and RST) depend on coreference chains of entities (Section 2), and only the global reading scheme takes advantage of the coreference pattern whereas the local reading breaks the chains. To find out whether coreference pattern is the key to the performance difference, we further ran a probe experiment where we read RST discourse relations in the order in which EDUs are arranged in the RST tree (i.e., left-to-right), and evaluated this model on novel-50 and IMDB62 with the same hyperparameter setting. The F1 scores turned out to be very close to the CNN2-DE (local) model, at 97.5 and 90.9. Based on this finding, we tentatively confirm the importance of the coreference pattern, and intend to further investigate how exactly it matters for the classification performance.

GR vs. RST. RST features in general give higher performance gains than GR features (Table 7). The RST parser produces a tree of discourse relations for the input text, thus introducing a "global view." The GR features, on the other hand, are more restricted to a "local view" on entities between consecutive sentences. While a deeper empirical investigation is needed, one can intuitively imagine that identifying authorship by focusing on the local transitions between grammatical re-

lations (as in GR) is more difficult than observing how the entire text is organized (as in RST).[12]

Error analysis. We conducted a brief error analysis in an effort to understand why discourse helps. Comparing performance by author, we found the least-represented author (Ambrose Bierce) obtains the biggest improvement from discourse. We speculate that although the document must be a certain length for discourse to "kick in", these features are effective even with few training examples. On the other hand, inspecting the gradients of the character bigrams for these cases reveals a higher incidence of 0s, suggesting the bigram feature is not as robust in the smaller sample space. We further note that two other authors who gained large improvements from the discourse features wrote a variety of genres (e.g., both supernatural/horror fiction and love stories), which we speculate manifests itself in different vocabularies that don't generalize well in character bigrams, but do have similar rhetorical styles which the discourse features can exploit.

6 Conclusion

We have conducted an in-depth investigation of techniques that (i) featurize discourse information, and (ii) effectively integrate discourse features into the state-of-the-art character-bigram CNN classifier for AA. Beyond confirming the overall superiority of RST features over GR features in larger and more difficult datasets, we present a discourse embedding technique that is unavailable for previously proposed discourse-enhanced models. The new technique enabled us to push the envelope of the current performance ceiling by a large margin.

Admittedly, in using the RST features with entity-grids, we lose the valuable RST tree structure. In future work, we intend to adopt more sophisticated methods such as RecNN, as per Ji and Smith (2017), to retain more information from the RST trees while reducing the parameter size. Further, we aim to understand how discourse embeddings contribute to AA tasks, and find alternatives to coreference chains for shorter texts.

[12]Note that, however, it is simpler to extract GR features, as we rely solely on a high-performance dependency parser, which is widely available, whereas for RST features, we need gold RST-labeled training data, which incurs higher cost and potentially relatively limited generalizability.

Acknowledgments

We thank Stephen Roller and three anonymous reviewers for their valuable feedback. Thanks to Sebastian Ruder for kindly offering his datasets in the exploratory stage of the project, and to Prasha Shrestha for sharing the code from her paper. The first author was supported by an NSF-GRFP fellowship.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1).

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Vanessa Wei Feng. 2015. *RST-style discourse parsing and its applications in discourse analysis*. Ph.D. thesis, University of Toronto.

Vanessa Wei Feng and Graeme Hirst. 2014. Patterns of local discourse coherence as a feature for authorship attribution. *Literary and Linguistic Computing*, 29(2):191–198.

Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2).

Alexander Hogenboom, Flavius Frasincar, Franciska De Jong, and Uzay Kaymak. 2015. Using rhetorical structure in sentiment analysis. *Communications of the ACM*, 58(7):69–77.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24. Association for Computational Linguistics.

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. In *CoRR: arXiv1412.6980*.

L.J.P. van der Maaten and G.E. Hinton. 2008. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9.

William Mann and Sandra Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Sebastian Ruder, Parsa Ghaffari, and John Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *CoRR, arXiv:1609.06686*.

Yunita Sari, Andreas Vlachos, and Mark Stevenson. 2017. Continuous n-gram representations for authorship attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 267–273. Association for Computational Linguistics.

Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891. Association for Computational Linguistics.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2011. Authorship attribution with latent dirichlet allocation.

Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, pages 155–159.

Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Patrick Juola, Aurelio López-López, Martin Potthast, and Benno Stein. 2015. *Overview of the Author Identification Task at PAN 2015*. Linda Cappellato and Nicola Ferro and Gareth Jones and Eric San Juan (eds.), CLEF 2015 Labs and Workshops, Toulouse, France.

Michael Strube and Udo Hahn. 1999. Functional centering grounding referential coherence in information structure. *Computational Linguistics*, 25(3).

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566. Association for Computational Linguistics.

Dat Tien Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330. Association for Computational Linguistics.