

Does Linguistic Structure Enrichment Enhance Coherence Assessment? Not With Current Architectures

Victor Mazzotti¹, Luiz Pereira², Marina Bitencourt dos Santos³, Helena Maia², Carlos Caetano²,
Nádia Felix⁴, and Sandra Avila²

¹ Instituto de Matemática, Estatística e Computação Científica (IMECC)

² Instituto de Computação (IC)

³ Instituto de Estudos da Linguagem (IEL)

Universidade Estadual de Campinas (UNICAMP), Campinas, SP, Brasil

⁴ Instituto de Informática (INF), Universidade Federal de Goiás (UFG), Goiânia, GO, Brasil

Abstract. Recent advances in large language models have transformed human–computer interaction. Despite their fluency, these models often produce texts that are grammatically correct but semantically incoherent, containing contradictions or disruptions in logical flow. This work investigates whether enriching text with syntactic and rhetorical information can improve incoherence prediction. Our experiments and analysis show that plain texts achieved higher accuracy because the added information was structurally and syntactically incompatible with the language model’s architecture. Additionally, to demonstrate the practical importance of coherence assessment, we performed zero-shot experiments on a Brazilian disinformation dataset, suggesting that textual coherence can serve as a proxy for detecting misleading content. Code and models are available at github.com/ittozzamV/cohereclassifier.

Keywords: coherence classification · story classification · rhetorical structure theory · part-of-speech

1 Introduction

The growing adoption of language models has transformed how computational systems interact with humans [17, 45], enabling the generation of more fluent, contextually rich text. These systems have evolved from simpler models, such as BERT [10], to large-scale architectures, such as the recent GPT-5 [29]. Driven by models with intuitive communication interfaces, such as ChatGPT [28] and Copilot [23], these systems enable realistic conversational interactions and support automatic content generation across diverse knowledge domains. Despite these impressive capabilities, a critical challenge remains: ensuring that the generated texts are textually coherent [41, 46].

Textual incoherence — characterized by internal contradictions, abrupt topic shifts, or flaws in logical structure — compromises the usefulness of these systems in sensitive applications, such as journalistic content production or education, where clarity and consistency are essential [11, 34, 40]. While traditional metrics for assessing grammatical correctness are well established in the literature [9, 25, 30], evaluating coherence remains a challenge, particularly in languages such as Portuguese, where annotated resources are scarce.

To address this challenge, we propose to investigate an approach that combines well-established linguistic theories, such as Rhetorical Structure Theory (RST) [22], which analyzes rhetorical relations between text segments, and grammatical classes commonly referred to as Part-Of-Speech (POS) [32], with machine learning techniques. Our goal is to understand whether enriching text with linguistic information can improve incoherence detection in automatically generated narratives, guiding further research in the field. To do so, we propose incorporating RST or POS information into the texts to enrich the classifier’s inputs.

Moreover, we evaluate, using a Brazilian disinformation dataset called FakeTrueBR [7], the hypothesis that textual incoherence can serve as an indirect indicator of disinformation. The main contributions of this work are:

- We propose a strategy to enrich texts with additional symbols derived from RST and POS information, and investigate whether it enhances model accuracy in detecting incoherence in stories.
- We show that coherence classification can serve as a *proxy* for detecting disinformation in offline scenarios, providing evidence of the practical applicability of coherence assessment. Our zero-shot cross-language pipeline also demonstrates that the knowledge can be reasonably transferred without further training on the new language.

2 Background and Related Work

In this section, we first revisit the RST, which provides the linguistic foundation for representing rhetorical relations in text. Then, we discuss POS tagging, which involves assigning grammatical class labels to each token in a text, providing syntactic information that complements the rhetorical structures considered in our approach. Finally, we review studies on coherence classification, focusing on approaches that integrate linguistic knowledge into language models to improve their ability to detect textual coherence.

Rhetorical Structure Theory (RST) [22] posits that, in addition to the explicit propositional content contained in the clauses of a text, there are

implicit propositions, called *relational propositions*, which emerge from the relations established between portions of the discourse. Skoufaki [37] demonstrates that it is possible to identify coherence breaks when a text is analyzed according to the definitions and diagrams of RST. Relational propositions result from combinations of clauses or text segments that, beyond their explicit propositional meaning, convey implicit content responsible for emphasizing or constraining certain aspects within a sentence [22]. Thus, these propositions are linked not only to text organization but also to relationships within its thematic structure, suggesting that applying this theory to text generation could guide overall organization and the selection of subsequent sentences.

In RST, a relation identifies the connection between two or more non-overlapping portions of text (Figure 1), called *Elementary Discourse Units* (EDUs), which can be classified as nuclei or satellites. Within these relations, nuclei represent the most salient part or the essential information of the relation, while satellites provide supporting or background information.

Part-Of-Speech (POS) Tagging involves assigning grammatical class labels to each token in a text (Figure 2). These labels identify the syntactic function of each word in a sentence, enabling a deeper analysis of the grammatical structure and meaning of the text [32]. POS tagging is often performed using supervised learning models, in which each word is assigned to its grammatical class. These models learn linguistic patterns from training data and apply them to predict POS labels for new unannotated texts.

Coherence Classification. The literature identifies two main categories for incorporating linguistic knowledge into a classifier: (i) embedding knowledge through additional layers or architectural modifications; and (ii) processing the input/output to include linguistic information for the generation and classification model.

Considering the first category, Ke et al. [15] propose combining POS information with the RoBERTa model by adding embedding layers that encode this

information for sentiment analysis. Linguistic embeddings condition text processing rather than being directly integrated into the input.

Abhishek [2] proposes four coherence-related tasks, including binary classification. Among the proposed models, the Vanilla Transformer relies on no extra linguistic information, and the Fact-aware Transformer combines text embeddings with linguistic ones (subject, verb, and object information) by adding an extra encoding layer. Both models are based on Longformer. In their experiments on the Grammarly Corpus of Discourse Coherence (GCDC) corpus [16], the Vanilla model outperforms the Fact-aware Transformer. Similarly, Liu and Strube [19] propose a fusion Transformer that combines text and linguistic embeddings for coherence classification. However, linguistic information comes from sentence-level relations given by the Penn Discourse TreeBank (PDTB). Unlike PDTB, RST enforces a hierarchical organization of relations. The authors evaluate their RoBERTa-based classifier with and without linguistic information, and find that, contrary to Abhishek [2], linguistic information improves GCDC performance. Liu et al. [18] use linguistic information (relations between nouns) to build sentence graphs that feed a Graph Convolutional Network to assess textual coherence.

The second category involves processing text to include linguistic information for the generation and classification models. Barzilay and Lapata [3] propose a coherence assessment approach by augmenting texts with syntactic annotations indicating whether nouns (entities) are subjects, objects, neither, or absent in a sentence. They use this information to construct grids. The authors argue that some grid patterns are more common in coherent texts and encode these patterns into vectors to feed a machine learning method. They employ a Support Vector Machine method for coherence assessment. Chernyavskiy [8] leverages RST relations as markers during GPT-2 fine-tuning to improve text generation. For that, they integrate special tokens <R> into raw texts, where R indicates an RST’s relation. Note that this requires only a minimal adjustment to the generative model to accept the special tokens, without any architectural modifications.

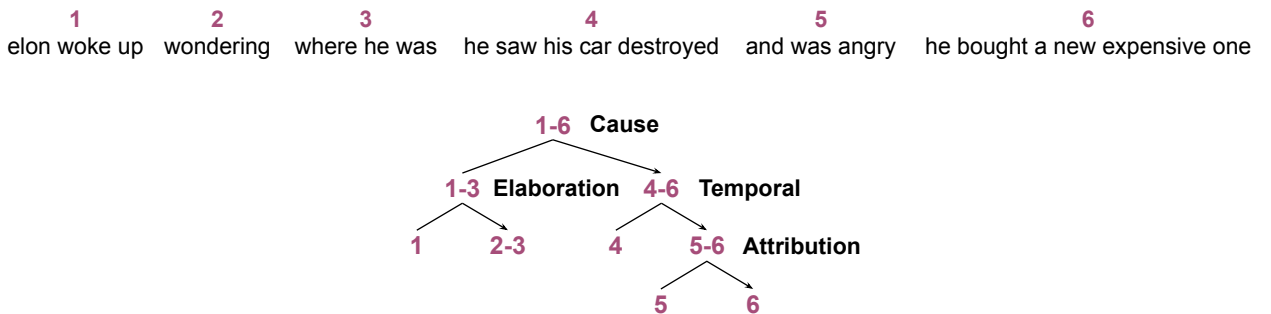


Fig. 1. Example of RST relations represented as a tree. The EDUs are shown in the first row. Each non-leaf node represents a relation, where the nucleus is the child node given by the tail of the arrow (left) and the satellite is the arrowhead (right). For instance, there is a Cause relation between EDUs 1-3 (nucleus) and EDUs 4-6 (satellite).

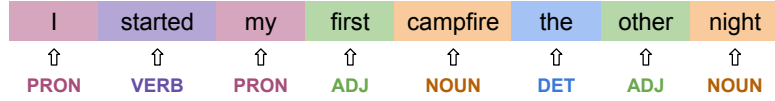


Fig. 2. Example of POS tags.

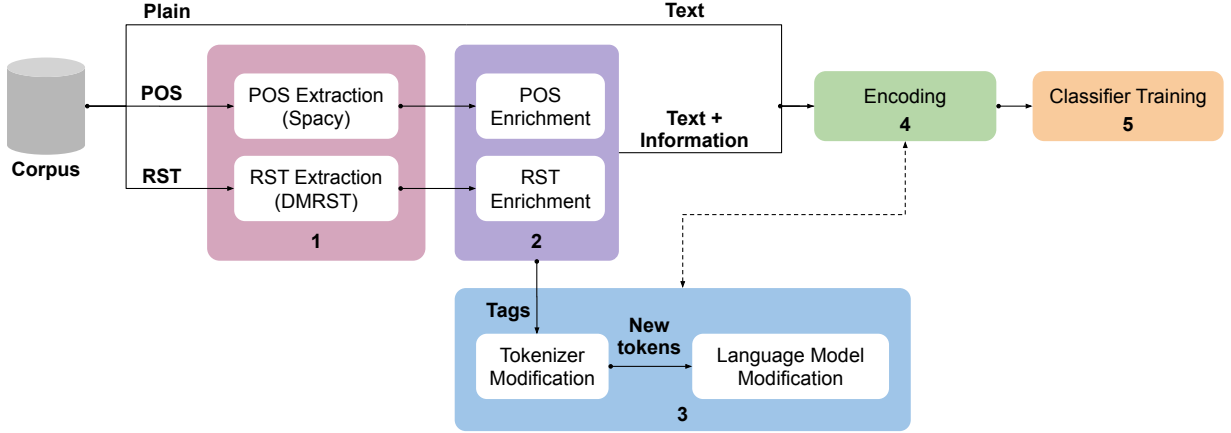


Fig. 3. Overview of the training pipeline of our coherence classifier in three distinct versions: Plain, POS, and RST. The first follows a two-step training process, while the last two use a five-step pipeline that integrates syntactic/rhetorical information.

Our proposed method falls into the second category. To the best of our knowledge, our work is the first to assess the impact of augmented inputs in Transformer-based models for coherence classification. In the first category, there is evidence both supporting the effectiveness of linguistic information [19] and suggesting that it brings no improvement [2]. Therefore, we aim to determine which of these two holds in the second category.

3 Coherence Classifier

Our proposed methodology, illustrated in Figure 3, consists of five steps: (1) extraction of POS/RST information; (2) text enrichment with POS/RST information; (3) tokenizer and language model modification; (4) text encoding; and (5) coherence classifier training. Steps (4) and (5) consist of the usual text classification steps. The pipeline version we call Plain includes only these two steps. Text encoding (4) is based on the XLM-RoBERTa Longformer model, including both the tokenizer and the text embedding generator. For (5), we add an extra layer to the XLM-RoBERTa Longformer model to classify the text as coherent or incoherent. We detail steps (1), (2), and (3), proposed in this work, in the following subsections.

3.1 Extraction of Linguistic Information

Extraction of RST Information. We employ the DMRST parser [21] to extract RST information. DMRST receives a text, separates it into tokens, and returns the EDU breaks and RST relations with nuclearity, requiring additional processing to obtain the EDUs. Figure 4 illustrates the extraction process.

Extraction of POS Information. To extract POS information, we use the spaCy parser [14]. This Python library provides high-accuracy and efficient POS tagging models. We used the parser to annotate the words in the texts with their respective grammatical classes, which were used for the POS enrichment stage. Figure 5 shows an example of this process.

3.2 Application of Text Enrichment

Although both RST and POS text enrichment consist of adding preprocessed information to plain text, they differ in how this information is added.

RST Enrichment. Enriching texts with RST consists of preprocessing texts and adding new symbols to language models. For this purpose, we create special tokens that represent RST relations, nuclearity, and EDU breaks. Each token consists of nuclearity and relation, separated by a colon (:), and enclosed by greater-than and less-than symbols (<>). The creation of these tokens and their positioning were inspired by the addition of special tokens to the Phi-3 model [1] to handle text excerpts that had some predefined formatting or structure. Since the RST parser has 19 relation categories and 2 nuclearity categories, 38 special tokens would be generated. However, in multinuclear relations (Joint, Same Unit, and Textual Organization), the nuclearity is the same for all participants considered as the nucleus. Thus, we generated only 35 tokens to represent all EDU separations, their relations, and their nuclearities. Table 1 shows a complete list of tokens. To represent EDU divisions, we add special tokens at the beginning and end of each EDU, depending on the RST relation. Although

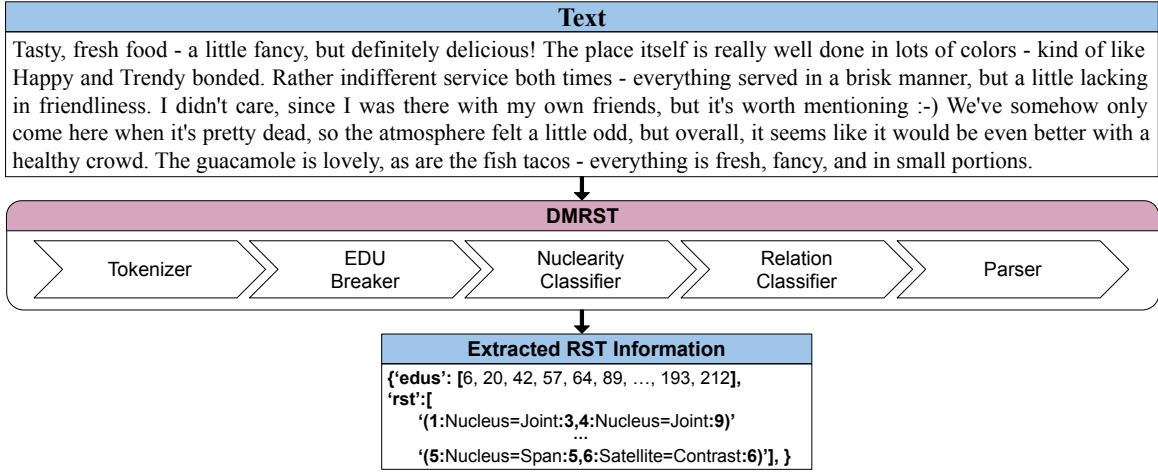


Fig. 4. Example of RST extraction using the DMRST parser.

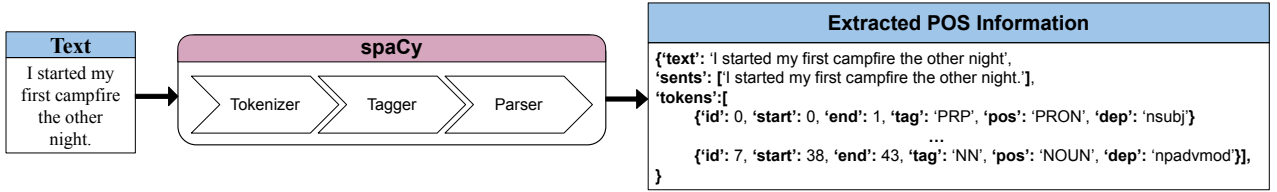


Fig. 5. Example of POS extraction using the spaCy library.

the order in which special tokens are added can influence the classification, we use the order returned by the parser, since there is no standard order. Table 2 shows the process of adding these tokens.

POS Enrichment. Enriching texts with POS is similar to enriching with RST, but the tokens only represent the grammatical classes of the words. We drew inspiration from the preprocessing presented by Wang et al. [43], which merges this information with the text, and adapted it to our context. Each token is formed by an underscore (_) followed by one of the 17 grammatical classes returned by the POS parser. The parts of speech used by the parser are the same as those defined in the Universal POS tags (UPOS) for treebank annotations in a multilingual setting [31]. Table 3 shows the complete list of tokens. The tokens are added after each word of the original text, with no space between them. Table 4 shows an example of adding special tokens.

3.3 Changing Tokenizer and Model

Tokenizer. To support the special tokens for the enrichment methods, new tokens must be added to the tokenizer’s vocabulary. Treating additional tokens as special allows them to be ignored during decoding and not split into subwords, while aiding understanding of the model’s results. Not splitting them into subwords is particularly important for the proposed methods, since splitting these tokens can alter their meaning and impair classification of long texts, as we may exceed the model’s limit by adding more tokens.

Model. The tokenizer update leads to a model update, since the model’s token embedding matrix must contain the same number of rows as the tokenizer’s vocabulary. The vectors added to the end of the embedding matrix are initialized randomly, impairing the model’s convergence. To alleviate this problem and allow the model to learn the vectors of the additional tokens more quickly, the weights of these vectors were replaced with values corresponding to the average of the weights of other vectors in the embedding matrix, a common practice when extending the vocabulary [13].

Loss Function. We used the weighted binary cross-entropy, an extension of binary cross-entropy that assigns different weights to each class, so that the minority class has a greater weight than the majority class (Equation 1). We chose this loss function because the GCDC corpus is imbalanced, with the coherent class as the majority and the incoherent class as the minority.

$$W = -\frac{1}{N} \sum_{i=1}^N [w_1 y_i \log(p_i) + w_0 (1 - y_i) \log(1 - p_i)], \quad (1)$$

where w_1 refers to the weight for the positive class (*i.e.*, incoherent texts), w_0 is the weight for the negative class (*i.e.*, coherent texts), $y_i \in \{0, 1\}$ is the binary label of example i , p_i is the probability predicted by the model for the positive class, and N is the total number of examples.

Table 1. Tokens generated for the RST annotation: one nucleus (N) and one satellite (S) token per mononuclear relation.

Relation	Token	Relation	Token
Topic Change	<N/S:Topic-Change>	Attribution	<N/S:Attribution>
Background	<N/S:Background>	Evaluation	<N/S:Evaluation>
Contrast	<N/S:Contrast>	Condition	<N/S:Condition>
Summary	<N/S:Summary>	Explanation	<N/S:Explanation>
Comparison	<N/S:Comparison>	Elaboration	<N/S:Elaboration>
Temporal	<N/S:Temporal>	Span	<N/S:Span>
Enablement	<N/S:Enablement>	Same Unit	<N:Same-Unit>
Cause	<N/S:Cause>	Joint	<N:Joint>
Topic Comment	<N/S:Topic-Comment>	Textual	<N:TextualOrganization>
Manner Means	<N/S:Manner-Means>	Organization	

Table 2. Example of plain text from which RST was extracted and the annotated text after the enrichment.

Plain Text	RST Enrichment
I chose to stay here because it's so close to the convention center for my son's bball tourny. The check in process was sooo slow. Even if I requested for a double bedroom a while back it was not given to us and then we have to change rooms the next day and the room they gave us smells so bad of cigarette smoke and it's supposed to be a non-smoking room!	<N:span><N:span>I chose to stay here<N:span><S:Explanation>because its so close to the convention center for my son's bball tourny. <N:span><S:Explanation><N:Joint>The check in process was sooo slow.<N:Joint><S:Contrast><N:Joint><N:Joint>Even if I requested for a double bedroom a while back<S:Contrast><N:Span>it was not given to us<N:Span><N:Joint><N:Joint><N:Joint>and then we have to change rooms the next day<N:Joint><N:Span><N:Same-Unit><N:Joint><N:Joint>and the room<N:Span><S:Elaboration>they gave us<N:Same-Unit><S:Elaboration><N:Same-Unit>smells so bad of cigarette smoke<N:Joint><N:Same-Unit><N:Joint>and it's supposed to be a non-smoking room!<N:Joint><N:Joint><N:Joint><N:Joint>

4 Experiments

4.1 Datasets

Grammarly Corpus of Discourse Coherence (GCDC). The GCDC corpus [16] contains English texts from four sources: Yahoo online forum posts, Yelp business reviews, emails from Hillary Clinton's office, and emails sent by Enron employees. Three specialists with previous annotation experience annotated each text. Each annotator classified the texts into three coherence levels: low, medium, or high. The coherence label was assigned to each text according to the most frequent label among the expert annotators. In case of disagreement among the annotators, the text was labeled with medium coherence.

We changed the corpus to a binary version, in which texts with high coherence were deemed coherent and those with low coherence were deemed incoherent. In our view, the presence of even one type of incoherence is enough to classify a text as incoherent, there being no middle ground for what might be considered medium coherence.

FakeTrueBR. The FakeTrueBR corpus [7] contains about 3500 Brazilian Portuguese texts from news articles, social media posts, and other formats of digital media, collected via a content crawler. The corpus is designed for disinformation and fact-checking analysis and stands out for pairing texts that show correspondence between disinformation and real news. We use it exclusively for zero-shot evaluation, without training or validation.

We evaluate our method on news texts (Brazilian Portuguese) based on our hypothesis that coherence-trained models can detect disinformation, given the presence of contradictions and informal language in these texts and in texts deemed incoherent.

4.2 Validation and Evaluation

We evaluate using balanced accuracy with mean and standard deviation across five runs, complemented by the Brier Score Loss [5], which assesses how well a model's predicted probabilities align with the observed outcomes. We report the results by corpus and its subsets.

5 Results

We used the XLM-RoBERTa model to train the coherence classifier on the GCDC corpus. We repeated each of the three proposed pipelines (Section 3) five times. We also conducted experiments with different numbers of scheduler cycles and different learning rates, but only for the Plain pipeline. We replicated the best values to the other pipelines to reduce the experiment load. We employed the best model on the English corpus on the Brazilian corpus.

5.1 GCDC

Table 5 presents balanced accuracy results for GCDC. First, we varied the number of cycles for learning rate (LR) restarts, keeping the LR at 5×10^{-5} . With 10 cycles, Plain exceeded 70.0% balanced accuracy, surpassing Abhishek's [2] accuracy result, which tends

Table 3. List of POS symbols generated by the spaCy library, their meaning, and the generated token.

UPOS	Meaning	Token	UPOS	Meaning	Token
SCONJ	Subordinate conjunction	_SCONJ	CCONJ	Coordinating conjunction	_CCONJ
NUM	Numeral	_NUM	PROPN	Proper noun	_PROPN
X	Others	_X	ADV	Adverb	_ADV
NOUN	Noun	_NOUN	ADJ	Adjective	_ADJ
PUNCT	Punctuation	_PUNCT	PRON	Pronoun	_PRON
AUX	Auxilliar	_AUX	SYM	Symbol	_SYM
ADP	Adposition	_ADP	PART	Particle	_PART
DET	Determinant	_DET	INTJ	Interjection	_INTJ
VERB	Verb	_VERB	SPACE	Space between words	None

Table 4. Example of plain text from which POS was extracted and the annotated text after the enrichment.

Plain Text	POS Enrichment
I chose to stay here because it's so close to the convention center for my son's bball tourny. The check in process was sooo slow. Even if I requested for a double bedroom a while back it was not given to us and then we have to change rooms the next day and the room they gave us smells so bad of cigarette smoke and it's supposed to be a non-smoking room!	I _PRON chose _VERB to _PART stay _VERB here _ADV because _SCONJ it _PRON 's _AUX so _ADV close _ADJ to _ADP the _DET convention _NOUN center _NOUN for _ADP my _PRON son _NOUN 's _PART bball _NOUN tourny _NOUN . _PUNCT The _DET check _NOUN in _ADP process _NOUN was _AUX sooo _ADV slow _ADJ . _PUNCT Even _ADV if _SCONJ I _PRON requested _VERB for _ADP a _DET double _ADJ bedroom _NOUN a _DET while _NOUN back _ADV it _PRON was _AUX not _PART given _VERB to _ADP us _PRON and _CCONJ then _ADV we _PRON have _VERB to _PART change _VERB rooms _NOUN the _DET next _ADJ day _NOUN and _CCONJ the _DET room _NOUN they _PRON gave _VERB us _PRON smells _VERB so _ADV bad _ADJ of _ADP cigarette _NOUN smoke _NOUN and _CCONJ it _PRON 's _AUX supposed _VERB to _PART be _AUX a _DET non _ADJ - _ADJ smoking _ADJ room _NOUN ! _PUNCT

to be higher than the balanced. We progressively reduced the number of cycles until reaching 1 to observe the classifier's behavior. Lastly, we reduced the LR as recommended by Godbole [12]. This last option had the highest average balanced accuracy across all pipelines. The Plain pipeline consistently outperformed the RST and POS, with RST ranking second.

We also analyze results by subset (Table 6) and hypothesize that the model performs differently across text types. Yelp consistently shows the worst performance, likely because its annotations tend to label colloquial writing as incoherent and norm-following text as coherent. Per-subset results also show that, although the RST pipeline outperforms the POS one overall, this does not hold for specific styles such as Clinton and Yahoo.

Table 7 presents the results for the Brier Score Loss. For this metric, we conducted experiments for each subset and for the entire corpus. This score ranges from 0 (best possible) to 1 (worst case). It indicates the model's confidence level when making correct or incorrect predictions. Unlike balanced accuracy, the POS pipeline performed best overall, while the Plain pipeline performed worst. Therefore, although the Plain pipeline achieved the best accuracy, it presented higher confidence levels even for incorrect predictions. Figure 6 illustrates this behavior.

5.2 FakeTrueBR

For the FakeTrueBR corpus, we carried out only zero-shot experiments, using models trained on the GCDC.

To compare results, we calculated the balanced accuracy (Table 8). The Plain pipeline performed best when trained on the GCDC corpus, followed by the RST pipeline. Although far from the average accuracy reported in the original paper (94.5%) [7], we achieved around 3/4 of the texts right in our best case while training models for other tasks and in other languages.

6 Discussion

GCDC. It may seem unexpected that enriched text underperforms plain text. In this subsection, we offer some justifications for this performance.

For POS enrichment, we understand that the type of information added is not well aligned with the objective of coherence evaluation, considering that incoherent texts can be grammatically correct, *i.e.*, the relationship between coherence and standard writing norms is not one of strict inclusion [35]. This may lead the model to generate false positives and false negatives, which explains an accuracy below that of the Plain pipeline.

On the other hand, Table 6 shows that for the Yahoo and Clinton subsets of the GCDC, the POS enrichment method performs better than the RST enrichment, which one would expect to be incorporating more relevant information for evaluating textual coherence. We understand that this is not due to a performance gain from the POS pipeline, but to poor performance from the RST one, due to structural in-

Table 5. Main results for the experiments performed on the GCDC corpus. Highlighted values represent the best result obtained, and underlined values represent the best score per pipeline. Cyc.: Cycles; LR: Learning Rate; Acc.: Accuracy.

Cyc.	Plain		Cyc.	RST		Cyc.	POS	
	LR	Acc. (%)		LR	Acc. (%)		LR	Acc. (%)
10	5×10^{-5}	71.2 ± 2.7	2	5×10^{-5}	67.1 ± 0.3	1	1×10^{-5}	<u>68.1 ± 0.4</u>
8	5×10^{-5}	72.0 ± 0.2	1	5×10^{-5}	67.2 ± 0.2			
5	5×10^{-5}	71.7 ± 0.2	1	1×10^{-5}	<u>69.0 ± 0.2</u>			
3	5×10^{-5}	71.5 ± 0.3						
2	5×10^{-5}	71.7 ± 0.2						
1	5×10^{-5}	71.1 ± 0.4						
1	1×10^{-5}	72.4 ± 0.1						

Table 6. Balanced accuracy for each pipeline with respect to each subset of the GCDC corpus. Highlighted values represent the best result, and underlined values represent the best score per pipeline.

Subset	Accuracy (%)		
	Plain	RST	POS
Clinton	73.1 ± 0.2	64.6 ± 0.5	67.7 ± 0.2
Yelp	64.5 ± 0.4	59.8 ± 0.3	59.1 ± 0.1
Enron	72.4 ± 0.2	<u>70.6 ± 0.2</u>	70.4 ± 0.1
Yahoo	74.1 ± 0.2	70.5 ± 0.5	<u>72.7 ± 0.3</u>

compatibility between the RST and the Transformer architecture.

From a mathematical perspective on the Transformer architecture, we argue that it is a non-isomorphic operator in the $\mathbb{R}^{d \times n}$ Euclidean space of sequences of n vectors of length d . Indeed, an isomorphism is a bijective operator. It has been shown that decoder only Transformers are injective, except on Lebesgue measure-zero subsets, when taken as discrete-to-continuous maps from prompts s in a finite vocabulary $\mathcal{V} \subset \mathbb{R}^{d \times n}$ to hidden states in $\mathbb{R}^{d \times n}$ [27]. In such a point-of-view, the Transformer cannot be surjective, as a map from a countable subset of $\mathbb{R}^{d \times n}$ cannot map to all uncountable elements of $\mathbb{R}^{d \times n}$ [36]. See the Appendix for more detail.

When visualizing the RST structure as a directed acyclic graph, the lack of isomorphism implies that the Transformer does not preserve the graph’s structure and relations. In this sense, the RST structure is flattened when passed through the Transformer, and the added information is retrieved as noise when training, thus the underperformance in comparison to the Plain pipeline.

FakeTrueBR. For the zero-shot experiments on the FakeTrueBR dataset, we verified that the Plain pipeline also performed better, achieving an accuracy comparable to its coherence assessment. In the context of domain transfer, we expected that the enriched pipelines would underperform the Plain one, as the latter performs better at assessing coherence. On the other hand, RST and POS pipelines performed even worse than their coherence counterparts, with POS enrichment being the worst of the two.

Table 7. Brier Score Loss for each subset of the GCDC corpus. The metric was calculated separately for each subset and globally, *i.e.*, using the entire corpus. Highlighted values represent the best result, and underlined values represent the best score per pipeline.

Subset	Brier Score Loss ↓		
	Plain	RST	POS
Clinton	0.300	0.272	0.236
Yelp	<u>0.296</u>	0.268	0.234
Enron	0.303	<u>0.263</u>	0.239
Yahoo	0.422	0.368	0.260
All	0.331	0.294	0.243

For POS enrichment, we understand that, as misleading and factually inaccurate texts are often syntactically well written to divert readers from their inaccuracy, the pipeline often predicts disinformation as real news. For RST enrichment, the performance is expected when taking it as noise for the model.

Nevertheless, the experiments with the FakeTrueBR corpus shows an underlying connection between textual incoherence and misleading content, as the Plain pipeline reached an accuracy of 3/4 of the dataset, despite being fine-tuned on another task and in another language.

7 Conclusion and Future Work

This paper investigated textual coherence classification in narratives by integrating RST and syntactic (POS) information into language models. Our proposed methodology combines linguistic theories with machine learning techniques. We showed that the plain-text model achieves the best accuracy compared with RST- and POS-enriched texts, and we provided mathematical and linguistic reasons for this. We also showed that expanding the model’s vocabulary with RST- and POS-derived symbols improved incoherence prediction confidence in some scenarios.

In this sense, we conclude that simply enriching texts with linguistic structure does not serve well the purpose of assessing coherence (at least not with RST and POS). Therefore, for future work, we intend to focus on incorporating linguistic features via additional embedding layers or other networks, such as Graph or Convolutional Neural Networks, which, as we discussed in Section 2, achieve superior results in text generation and classification. One way to advance in

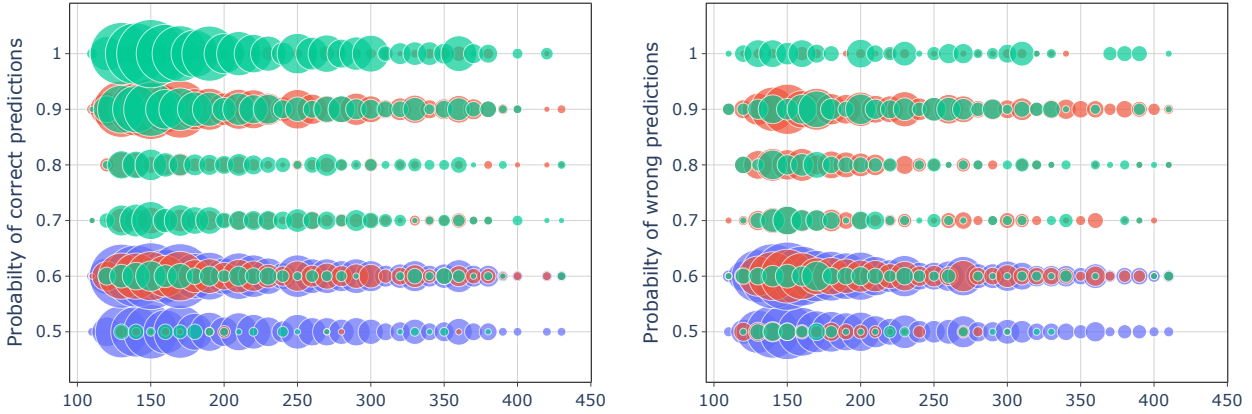


Fig. 6. Probability distribution of correct and incorrect predictions on the GCDC corpus with Vanilla ●, RST ●, and POS ● pipelines. The circles’ diameters are proportional to the number of texts corresponding to each length-probability pair.

Table 8. Obtained results with the balanced accuracy metric for the FakeTrueBR corpus in a zero-shot scenario. Highlighted values represent the best result, and underlined values represent the best score per pipeline.

Accuracy (%)		
Plain	RST	POS
73.1 ± 0.8	<u>61.5</u> ± 0.5	59.5 ± 0.6

this direction would be to replicate these strategies for RST. We also hypothesize that enriching text with RST would yield better results when coupled with architectural modifications that account for hierarchical structures, such as Poincaré Embeddings [26] and Graph-aware Isomorphic Attention [6].

Recent works [19, 20] also address the use of the PDTB [33, 44], a linear discourse annotation framework, and are therefore structurally aligned with the Transformer architecture. Thus, enriching text with PDTB, instead of RST or POS, could achieve better results. We intend to implement this enrichment and repeat the experiments, hypothesizing that its performance is directly related to the overall adequacy of the text enrichment technique.

On the other hand, experiments showed that the models not only serve for incoherence detection but also presented promising results in zero-shot tests on FakeTrueBR. However, the correlation between coherence and disinformation, while statistically relevant, demands further investigation to distinguish between intentionally misleading texts and those that are simply poorly structured. For the future, it would be relevant to replicate the zero-shot experiments in the same language domain on which the classifier was trained.

8 Ethics Statements and Limitations

This study adheres to the ethical principles of Natural Language Processing research. The proposed models are intended solely for academic research, involve no human subject data, and use only publicly available datasets. The following limitations and statements apply.

Annotated Data. Although our models are multilingual, the DMRST parser [21] supports only six languages: English, Portuguese, Spanish, German, Dutch, and Basque. This limitation arises not only from the parser itself but also from the scarcity of available annotated datasets. Annotating RST requires expert knowledge, and even trained annotators may produce inconsistent structures due to differences in interpretation or linguistic nuances.

Textual Coherence and Arguments. We define discourse as structured text composed of coherent sentences, where coherence (local, global, or topical) ensures semantic connectivity and non-contradiction. However, coherent texts may still contain biased, misleading, or factually incorrect content. For instance, some GCDC corpus examples express xenophobic views that are structurally coherent but incoherent with our worldview.

Carbon and Water Usage Estimator. This experiment consumed 385 kWh, with an estimated footprint of 37.86 kgCO₂e and 7,154.97 L of water in Brazil, as calculated with wAIter (beta version) [4]. Since we used three GPUs, the total estimated consumption was 1,155.00 kWh, corresponding to a footprint of 113.58 kgCO₂e and 21,464.91 L of water.

Generative AI and AI-assisted Technologies.

The authors occasionally made use of DeepL Translate, ChatGPT and Gemini during the preparation of this paper to assist with language refinement and the editing process. The authors carefully reviewed and revised all text, assuming full responsibility for the final content.

References

1. Abdin, M.I., Ade Jacobs, S., Awan, A.A., et al.: Phi-3 technical report: A highly capable language model locally on your phone. Tech. Rep. MSR-TR-2024-12 (2024)
2. Abhishek, T., Rawat, D., Gupta, M., Varma, V.: Transformer models for text coherence assessment. arXiv:2109.02176 (2021)
3. Barzilay, R., Lapata, M.: Modeling local coherence: An entity-based approach. In: ACL (2005)
4. Breder, G., Ferro, M.: wAIter: Serving awareness-environmental footprint estimator for artificial intelligence models (2025). <https://doi.org/10.5281/zenodo.17247278>
5. Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly weather review (1950)
6. Buehler, M.J.: Graph-aware isomorphic attention for adaptive dynamics in transformers. APL Machine Learning **3**(2), 026108 (2025). <https://doi.org/10.1063/5.0256873>
7. Chavarro, J.P., Carvalho, J.T., Portela, T.T., Silva, J.C.: FakeTrueBR: Um corpus brasileiro de notícias falsas. In: ERBD (2023)
8. Chernyavskiy, A., Ilvovsky, D., Galitsky, B.: Correcting texts generated by transformers using discourse features and web mining. In: RANLP (2021)
9. Choshen, L., Abend, O.: Automatic metric validation for grammatical error correction. In: ACL. pp. 1372–1382 (2018)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
11. Fatima, N., Imran, A.S., Kastrati, Z., Daudpota, S.M., Soomro, A., Shaikh, S.: A systematic literature review on text generation using deep neural network models. IEEE Access (2022)
12. Godbole, V., Dahl, G.E., Gilmer, J., Shallue, C.J., Nado, Z.: Deep learning tuning playbook (2023), http://github.com/google-research/tuning_playbook, v. 1.0
13. Hewitt, J.: Initializing new word embeddings for pre-trained language models. <https://nlp.stanford.edu/~johnehw/vocab-expansion.html> (2021)
14. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). <https://doi.org/10.5281/zenodo.1212303>
15. Ke, P., Ji, H., Liu, S., Zhu, X., Huang, M.: Sentilare: Sentiment-aware language representation learning with linguistic knowledge. arXiv:1911.02493 (2019)
16. Lai, A., Tetreault, J.: Discourse coherence in the wild: A dataset, evaluation and methods. In: SIGdial (2018)
17. Lee, M., Srivastava, M., Hardy, A., Thickstun, J., Durmus, E., Paranjape, A., et al.: Evaluating human-language model interaction. TMLR (2023)
18. Liu, W., Fu, X., Strube, M.: Modeling structural similarities between documents for coherence assessment with graph convolutional networks. In: ACL (2023)
19. Liu, W., Strube, M.: Discourse relation-enhanced neural coherence modeling. In: ACL (2025)
20. Liu, W., Strube, M.: Joint modeling of entities and discourse relations for coherence assessment. In: EMNLP (2025)
21. Liu, Z., Shi, K., Chen, N.: DMRST: A joint framework for document-level multilingual rst discourse segmentation and parsing. In: CODI. pp. 154–164 (2021)
22. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text **8**(3), 243–281 (1988)
23. Microsoft: Copilot. <https://news.microsoft.com/september-2023-event/> (2023)
24. Munkres, J.R.: Analysis on Manifolds. CRC Press, Boca Raton (2018)
25. Napoles, C., Sakaguchi, K., Post, M., Tetreault, J.: Ground truth for grammatical error correction metrics. In: ACL. pp. 588–593 (2015)
26. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
27. Nikolaou, G., Mencattini, T., Crisostomi, D., Santilli, A., Panagakis, Y., Rodolà, E.: Language models are injective and hence invertible. In: ICLR (2026)
28. OpenAI: ChatGPT. <https://openai.com/index/chatgpt>
29. OpenAI: GPT-5 system card. Tech. rep. (2025), <https://cdn.openai.com/gpt-5-system-card.pdf>
30. Park, C., Yang, Y., Lee, C., Lim, H.: Comparison of the evaluation metrics for neural grammatical error correction with overcorrection. IEEE Access **8** (2020)
31. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: LREC (2012)
32. Pradhan, S., Ward, W., Hacioglu, K., Martin, J.H., Jurafsky, D.: Semantic role labeling using different syntactic views. In: ACL. pp. 581–588 (2005)
33. Prasad, R., Dinesh, N., Lee, A., Miltasakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse TreeBank 2.0. In: LREC (2008)
34. Redeker, G.: Coherence and structure in text and discourse. Abduction, belief and context in dialogue **233**(263) (2000)
35. Reinhart, T.: Conditions for text coherence. Poetics Today (1980)
36. Rudin, W.: Principles of Mathematical Analysis. McGraw-Hill, New York, 3rd edn. (1976)
37. Skoufaki, S.: Rhetorical structure theory and coherence break identification. Text & Talk (2020)
38. Sommerville, D.M.Y.: An Introduction to the Geometry of N Dimensions. E. P. Dutton and Company, New York (1929), reprinted by Dover Publications, 1958
39. Thickstun, J.: The transformer model in equations. University of Washington: Seattle, WA, USA (2021)
40. Thompson, I.: Readability beyond the sentence: Global coherence and ease of comprehension. JTWC **16**(1), 131–140 (1986)
41. Vakulenko, S., de Rijke, M., Cochez, M., Savenkov, V., Polleres, A.: Measuring semantic coherence of a conversation. In: ISWC. pp. 634–651 (2018)
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
43. Wang, F., Ding, L., Rao, J., Liu, Y., Shen, L., Ding, C.: Can linguistic knowledge improve multimodal alignment in vision-language pretraining? TOMM **20** (2024)
44. Webber, B., Prasad, R., Lee, A., Joshi, A.: The Penn Discourse Treebank 3.0 Annotation Manual (2019)
45. Zhang, C., Chen, J., Li, J., Peng, Y., Mao, Z.: Large language models for human-robot interaction: A review. BIROB (2023)
46. Zhao, Q., Niu, J., Liu, X., He, W., Tang, S.: Generation of coherent multi-sentence texts with a coherence mechanism. CSL (2023)

Appendix

In a continuous theoretical setting, such as proposed in the original Transformers paper [42], a decoder-only Transformer with L layers is the composition of L Transformer-blocks [39]. A Transformer block is a parametrized function $f_\theta : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^{d \times n}$. For an input $x = (x_1, x_2, \dots, x_{d \times n}) \in \mathbb{R}^{d \times n}$, $f_\theta(x) = z$ is calculated as follows: the Query (Q), Key (K) and Value (V) matrices are trainable weights over the input:

$$Q^{(h)}(x_i) = W_{h,q}^T x_i, \quad K^{(h)}(x_i) = W_{h,k}^T x_i, \quad V^{(h)}(x_i) = W_{h,v}^T x_i,$$

for each head h and trainable weight matrices W . Then, the self-attention between vectors x_i and x_j is defined as

$$\alpha_{i,j}^{(h)} = \text{softmax} \left(\frac{\langle Q^{(h)}(x_i), K^{(h)}(x_j) \rangle}{\sqrt{k}} \right).$$

Then, the self-attention is summed over weights per head

$$u'_i = \sum_{h=0}^H W_{c,h}^T \sum_{j=1}^n \alpha_{i,j}^{(h)} V^{(h)}(x_j),$$

and normalized

$$u_i = \text{LayerNorm}(x_i + u'_i; \gamma_1, \beta_1).$$

This vector goes through an activation function (such as ReLU):

$$z'_i = W_2^T \text{ReLU}(W_1^T u_i),$$

which is again normalized to obtain the output of the Transformer block

$$z_i = \text{LayerNorm}(u_i + z'_i; \gamma_2, \beta_2).$$

The LayerNorm function is defined as

$$\begin{aligned} \text{LayerNorm}(z; \gamma, \beta) &= \gamma \frac{(z - \mu_z)}{\sigma} + \beta, \\ \mu_z &= \frac{1}{k} \sum_{i=0}^k z_i, \\ \sigma &= \sqrt{\frac{1}{k} \sum_{i=1}^k (z_i - \mu_z)^2}. \end{aligned}$$

The Transformer is then the composition of L Transformer blocks, $f_{\theta_L} \circ \dots \circ f_{\theta_1} \in \mathbb{R}^{d \times n}$.

In these settings, we aim to prove the following theorem:

Theorem. Let $f_\theta : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^{d \times n}$ be the continuous decoder-only Transformer mapping sequences of n continuous token embeddings of dimension d to their final hidden states. Then, f is not surjective.

Proof. Our proof consists in decomposing the Transformer f as $f = \phi \circ g$, where g represents the composition of all architectural components before the final normalization function, represented by ϕ . It suffices to prove that the outermost function ϕ is not surjective to prove that f is not surjective.

For the LayerNorm, the normalized vector $z = \frac{x - \mu}{\sigma}$ is such that it lies on a hyperplane orthogonal to the identity vector, that is, $\sum_{i=1}^d z_i = 0$. The normalized vector also lies on a hypersphere of radius $\sqrt{d}$, that is, $\|z\|_2^2 = d$, where $\|\cdot\|_2$ is the euclidean norm. Then, the normalized vector lies on the manifold M_i , which is the intersection of a hyperplane with a hypersphere. Thus, the manifold where the normalized vector lies has dimension $d - 2$ [38].

Since the map ϕ applies individually to the entire sequence of n vectors, the full image space manifold is the Cartesian product of each manifold M_i :

$$M = M_1 \times M_2 \times \dots \times M_n.$$

As such, the dimension of the image is

$$\dim(M) = \sum_{i=0}^n \dim(M_i).$$

Since each M_i has dimension $d - 2$, $\dim(M) = n(d - 2) = nd - 2n$. Because in any practical Transformer $n \geq 1$ and $d \geq 2$, we have

$$\dim(M) = nd - 2n < nd = \dim(\mathbb{R}^{d \times n}).$$

Therefore, the image of f , denoted $\text{Im}(f) = \text{Im}(\phi \circ g) \subset \text{Im}(\phi) \subset M$, is constrained to a submanifold whose dimension is less than the dimension of the codomain. By properties of Lebesgue measure, any smooth manifold $M \subset \mathbb{R}^n$ of dimension $m < n$ has a Lebesgue measure of zero in $\mathbb{R}^n$ [24]. As such, $\text{Im}(f)$ has measure zero in $\mathbb{R}^{d \times n}$, implying that $\text{Im}(f)$ is a proper subset of $\mathbb{R}^{d \times n}$, and f is not surjective. $\square$

On the other hand, when taking the Transformer as a discrete-to-continuous operator, Cantor's Theorem states that the operator cannot be surjective [36]. An operator as such is the one presented in [27], defined by $f : \mathcal{V}^{\leq K} \times \mathbb{R}^p \rightarrow \Delta^{|\mathcal{V}|-1}$, that is, an operator from the finite vocabulary set with p parameters to the simplex of dimension $|\mathcal{V}| - 1$. Since the simplex is an uncountable subset of $\mathbb{R}^n$, the map f cannot be surjective and, thus, it is not an isomorphism.