Automatic Annotation of Direct Speech in Written French Narratives

Noé Durandard
Deezer Research & EPFL
noe.durandard@epfl.ch

Viet-Anh Tran
Deezer Research
vatran@deezer.com

Gaspard Michel
Deezer Research
gmichel@deezer.com

Elena V. Epure
Deezer Research
eepure@deezer.com

Abstract

The automatic annotation of direct speech (AADS) in written text has been often used in computational narrative understanding. Methods based on either rules or deep neural networks have been explored, in particular for English or German languages. Yet, for French, our target language, not many works exist. Our goal is to create a unified framework to design and evaluate AADS models in French. For this, we consolidated the largest-to-date French narrative dataset annotated with DS per word; we adapted various baselines for sequence labelling or from AADS in other languages; and we designed and conducted an extensive evaluation focused on generalisation. Results show that the task still requires substantial efforts and emphasise characteristics of each baseline. Although this framework could be improved, it is a step further to encourage more research on the topic.

Le pauvre garçon voulut paraître fort, et il répéta plusieurs fois :

[**– Oui..., du courage !**]$_{DS}$

[**– Eh bien,**]$_{DS}$ s'écria le bonhomme, [**j'en aurai, nom d'un tonnerre de Dieu ! Je m'en vas la conduire jusqu'au bout.**]$_{DS}$

The poor fellow tried to show himself brave, and repeated several times. [**"Yes! courage!"**]$_{DS}$

[**"Oh,"**]$_{DS}$ cried the old man, [**"so I will have, by God! I'll go along o' her to the end!"**]$_{DS}$

Figure 1: Excerpts of *Madame Bovary* by Gustave Flaubert (1856). Translation by Eleanor Marx-Aveling.

1 Introduction

Prose fiction makes whole worlds emerge. Authors make use of different strategies to create narratives and convey the *storyworld*. Novels intertwine narrators' words to build the atmosphere and tell the story, with words stemming from characters inhabiting the fictive world that disclose their personality and depict them directly via dialogues or direct speech (DS) (James, 2011; Hühn et al., 2014).

The development of algorithms to perform the automatic annotation of direct speech (AADS) in written text has been of high interest for literary studies. This task consists in retrieving lines uttered by the characters of a narrative in contrast to words delivered by the narrator of the story. One goal of AADS has been to compare fiction works by different authors or stemming from different genres or time periods. DS was then studied as a literary device carrying specific purposes and disclosing compelling cultural information (Muzny et al., 2017; Egbert and Mahlberg, 2020). AADS is also central in narrative understanding endeavors. DS has been then considered as the main realisation of characters, their means to gain volume and depth, and come alive to the readers. In this context, AADS is often regarded as a pre-processing step that enables downstream analysis such as DS speaker attribution (Cuesta-Lazaro et al., 2022), that can in turn serve to assemble characters networks (Labatut and Bost, 2019), or model personas (Sang et al., 2022).

AADS has been widely performed for English literature, leveraging strict formatting conventions (e.g. quotes or long dashes) to extract DS through simple regular expression—regex (Bamman et al., 2014; O'Keefe et al., 2012; Elson and McKeown, 2010). Yet, in other languages, dialogues may be less strictly segregated from narration and typographic conventions can be more flexible. Hence, more complex solutions based on lexical features have been developed, mainly for German (Brunner, 2013; Jannidis et al., 2018; Brunner et al., 2020). These lexical features were either manually defined and exploited with classical machine learning algorithms such as Random Forest (Brunner, 2013), or were inferred indirectly from text in deep learning frameworks (Jannidis et al., 2018; Brunner et al., 2020) using Recurrent Neural Networks or lan-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7129–7147
July 9-14, 2023 ©2023 Association for Computational Linguistics

guage models such as BERT (Devlin et al., 2019).

For other languages, including French, there are very few AADS efforts. Schöch et al. (2016) propose *Straight Talk!*, a corpus of 40 chapters from 19th century French novels annotated per sentence if containing DS or not, and performed binary classification using 81 engineered features. The corpus was quite large, but sentences were poorly segmented with a high impact on results; annotations did not consider *incises* (i.e. narrative breaks within the same DS turn as in Figure 1); despite a high overall F1-score (93%), some writing styles were very challenging (for instance in homodiegetic narratives, where the narrator is a fully fledged character in the storyworld and may relate the story at the first person). In another work, Sini et al. (2018a) adopted a feature engineering approach as well. They combined it with rules to segment and identify paragraphs containing DS, and then to extract *incises* from mixed paragraphs. Still, the method was tested on a small corpus, a subset of *SynPaFlex* (Sini et al., 2018b) with excerpts from only two novels. Finally, Byszuk et al. (2020) considered AADS in multilingual settings using BERT, but on an even smaller French corpus.

The goal of the current work is to create an unified framework for designing and evaluating AADS models in French, which in return we hope to encourage more research on the topic[1]. Specifically, we address existing limitations on multiple fronts:

1. We catalogued and consolidated the largest-to-date dataset of French narratives manually annotated with DS tags at the word level based on 4 existing corpora. First, we re-annotated *Straight Talk!* (Schöch et al., 2016) to reach a finer granularity: from sentence to word level. Second, we extended the *SynPaFlex* (Sini et al., 2018a) sparse annotations, initially done on chapter excerpts, to cover the whole chapters. We also incorporated two new corpora as they were: *fr-LitBank*, the French variant of the Multilingual BookNLP project (Lattice, 2022) and an extension of *SynPaFlex* (Sini et al., 2018a) provided by the authors. Our dataset is made of 86 whole chapters ($680K$ annotated tokens), extracted from French novels published during the 19th and 20th centuries.

2. We modelled AADS as a token classification task, which we argue as more suitable for *incises* identification. This approach allowed us to benchmark state-of-the-art sequence labelling models such as French fine-tuned transformers (Martin et al., 2020) for the first time for AADS. We also re-implemented the most popular AADS baselines from other languages to fit French language peculiarities and trained them on our dataset. In our selection, we included baselines that did not require extensive manual feature engineering to encourage generalisation over various writing styles.

3. We devised an extensive evaluation covering text with varied formatting quality. Apart from traditional token- and span-level strict precision, recall and F1-score metrics (Yadav and Bethard, 2018), we adapted ZoneMap (Galibert et al., 2014), a metric stemming from page segmentation literature, to our task. This allowed us to quantify the effect of various error types made by the models and deepen our understanding of their limitations.

Results show that rule-based baselines using regular expressions remain a good choice when texts are well-formatted. Deep learning solutions are however more effective and achieve satisfactory results even on narratives with poor formatting quality. Their most common issue is that they still miss to catch whole DS sequences. We also conducted a qualitative analysis to bring insights on the strengths and weaknesses of various models, and defined the directions for future endeavors.

2 Literature Review

We further review AADS solutions for any language.

2.1 Rule-based AADS

While conventions may vary across languages and novels, DS tends to be enclosed within quotation marks (e.g. «...»; "..."), or introduced with long dashes (e.g. —...; –...). Regarded as pre-processing, simple AADS methods relying on regex with low computational costs are favored (Thomas, 2012; Cunha and Arabyan, 2004). The AADS module of BookNLP (Bamman et al., 2014), the reference pipeline developed for computational narrative understanding in English, first determines

[1]Code and data are publicly available at https://github.com/deezer/aads_french

the most used quotation mark type from a predefined set; then it tags every passage in between the selected quotation mark pair as DS. This yields performances around an F1-score of 90% when evaluated as a token-level binary classification task on the LitBank 19th century book corpus (Sims and Bamman, 2020). Variations of this approach considering more quotation mark types than the most used one are also common (Cuesta-Lazaro et al., 2022; Yoder et al., 2021; Byszuk et al., 2020; O'Keefe et al., 2012). Almost perfect F1-scores ($96 - 99\%$) are then reported on various English corpora.

However, when working with heterogeneous corpora, texts with poorer encoding quality (because of Optical Character Recognition errors or changing editing standards over time), or other languages, typographic AADS appears to be limited (Byszuk et al., 2020; Muzny et al., 2017). For instance, performances on French and German decrease to a F1-score of 92% and down to 65% for Norwegian (Byszuk et al., 2020). Similarly, we observe the F1-score decreasing to 77% on a more challenging English corpus (Muzny et al., 2017).

To overcome these issues, more complex rule-based systems that leverage semantic and syntactic cues besides typographic markers have been proposed for English (Muzny et al., 2017) and German (Tu et al., 2019). Empirical studies revealing writing style differences between DS and narration (Egbert and Mahlberg, 2020) have supported this direction. The lack of DS markers and the prevalence of *incises* in French literature has also led Sini et al. (2018a) to devise more sophisticated regex based on dependency parsing and Part-of-Speech (POS) tags, yielding an F1-score of 89.1%.

2.2 Machine Learning-based AADS

With an increasing availability of annotated corpora, AADS based on machine learning has been explored more and more, in particular on German literature (Brunner, 2013; Tu et al., 2019; Brunner et al., 2020). Works on other languages, such as French (Schöch et al., 2016) or Swedish (Ek and Wirén, 2019), have also emerged, while remaining sparse and isolated. ELTeC multilingual initiative (Odebrecht et al., 2021) has encouraged the investigation of multilingual approaches too (Byszuk et al., 2020; Kurfalı and Wirén, 2020).

All these endeavors exploit syntactic and semantic features of DS segments beyond typographic cues, either through feature engineering or by learning features from text with end-to-end deep learning. Brunner (2013) trained a Random Forest on 80 syntactic and semantic features extracted at the sentence level from a corpus of 13 short German narratives. Her method showed a 3 point improvement compared to the rule-based AADS baseline, though with a large standard deviation (19%). This approach was later adapted to French by Schöch et al. (2016) on a corpus of 40 book chapters.

In the recent years, the successful application of deep learning to a wide-range of NLP tasks has led to the adoption of these models for AADS too. Brunner et al. (2020) proposed to use a BiLSTM-CRF (Huang et al., 2015) on text encoded with Flair (Akbik et al., 2018), FastText (Mikolov et al., 2018) and a multilingual BERT (Devlin et al., 2019), as well as to fine-tune the German-language BERT (Chan et al., 2020) for AADS on German narratives. Byszuk et al. (2020) fine-tuned a multilingual BERT and reported an overall F1-score of 87.3% at the token level. However, the score per language is missing, making it challenging to assess the benefits of the approach for individual cases. Kurfalı and Wirén (2020) adopt a zero-shot framework and remove DS typographic markers from the test corpora. They trained a multilingual BERT on silver-labelled data obtained with regex AADS and report token-level F1-score of 85% on English, 73% on Swedish and 64% on German.

In summary, research dedicated to French remains very sparse and suffers from a lack of comparability because of differences among the studied corpora, task modeling focuses (token vs. sentence classification), or imposed research scenario (without typographic markers, multilingual, zero-shot).

3 French Narrative Corpora for AADS

We consolidate a large dataset of French novel excerpts, manually annotated with DS labels at the word level. Built upon existing endeavors, the final dataset is a compilation of four sub-corpora, individually referred to as *Straight Talk!* (*ST!*) (Schöch et al., 2016), *SynPaFlex* (*SPF*) (Sini et al., 2018a), an extension of *SynPaFlex* provided to us by the authors (*SB*), and *fr-LitBank* (*fr-LB*) (Lattice, 2022). While *fr-LB*, *SPF*, and *SB* have overall good encoding and segmentation quality, *ST!* is poorly formatted with some files lacking line breaks, for instance.

Each sub-corpus contains French novels from

	#Fs	#Toks	#Sents	DS%(std)
Train	37	333,638	22,917	40 (30)
Valid	6	67,568	5,332	34 (19)
Test$_C$	6	59,016	3,803	29 (19)
Test$_N$	37	222,650	14,406	37 (21)

Table 1: Number of files (#Fs), tokens (#Toks), and sentences (#Sents), and % of DS tokens (with standard deviation) for train, validation, clean test (Test$_C$) and noisy test (Test$_N$) splits.

public-domain published between 1830 and 1937[2]. It results in an aggregated corpus gathering 86 chapters extracted from 44 novels. The full dataset comprises more than $680K$ words, 8826 DS spans which represent 37% of the total tokens. However, we can observe large variations of DS presence across files (see Appendix A), from no DS in the excerpt named *madame_bovary_première_9* to 92% of the words being labelled as DS in *mystères_de_paris_2_troisième_16*. Appendix A shows the excerpts and more dataset details.

The sub-corpora, *fr-LB*, and *SB*, were kept in the form provided by the original works. In contrast, we modified the ground-truth annotations of *ST!* and *SPF* in order to align them with the other two sub-corpora and the adopted approach to model the problem—binary classification at the token level– and to exhaustively cover chapters, not only excerpts. In particular, *ST!* annotations lacked granularity as text segments were labelled as Narration or Mixed (comprising both Narration and DS), so we corrected those. As for *SPF*, the annotations were very sparse among the 27 chapters; hence we extended them to whole chapters.

The re-annotation process was mainly led by one author using Doccano (Nakayama et al., 2018). A selection of 5 files were doubly annotated by a co-author to check labeling reliability. The obtained pairwise Cohen's κ score (Cohen, 1960) was 97%, which is considered almost perfect agreement. The re-annotated dataset is shared with the code.

The dataset is then split into train, validation and test sets. The files from the three well-formatted sub-corpora (*fr-LB*, *SPF*, *SB*) are randomly divided in order to ensure a proportion of $0.8/0.1/0.1$ for train, validation, and test, respectively, and that at least one file from each sub-corpus can be found in each split. Each file can be found in only one split, but we sometimes have files from the same novel present in all splits, especially those originating from the *SPF* sub-corpus (*Les Mystères de Paris* by Eugène Sue and *Madame Bovary* by Gustave Flaubert). Finally, *ST!* is kept for test only as a challenge dataset. Indeed, contrary to the other sub-corpora mentioned above, this latter sub-corpus suffers from largely unequal formatting quality across files. Some chapters are completely devoid of line break which makes them, wrongly, appear as one unique paragraph, while others exhibit misplaced line breaks, sometimes in the middle of sentences. *ST!*'s formatting peculiarities make it a good test for generalisation, especially on noisier text. This challenging set is also referred to as a noisy test set (Test$_N$) in contrast to the clean test set (Test$_C$) stemming from the split of the three well-formatted sub-corpora —that are also used for training and validation.

Dataset statistics are shown in Table 1. More details on split composition in terms of files can be found in Appendix A.

4 Methods

Popular baselines from the two AADS approaches (rule-based and machine learning-based), including those designed for other languages, were modified to fit the characteristics of French. AADS was then formalized either as a text matching and extraction task, when using regex, or as a sequence labelling task, when using deep learning models. For the latter, the AADS models returned a binary label per token, (*O / DS*) as in other related works (Brunner et al., 2020; Ek and Wirén, 2019; Jannidis et al., 2018). While regex has been more common, to our knowledge, this is the most extensive attempt to explore deep learning for AADS in French narratives.

4.1 Rule-based AADS Baselines

We adapted two rule-based systems (Byszuk et al., 2020; Bamman et al., 2014) for our framework.

Byszuk et al. (2020) compiled a list of various quotation marks and dashes used to introduce characters' DS, which we kept the same. However, we modified the definition of paragraphs, the input to the regex system, to be spans of text until a break line. Regular expressions were after applied, as they were, to extract the text enclosed by quotation marks or introduced by a dialogue dash.

In contrast, Bamman et al. (2014)'s method was driven by the hypothesis that each text used a single typographic convention for DS. Thus, they identified the most used quotation mark in the analyzed

[2]This period is chosen because it is copyright-free.

document from a predefined list. Then, regex was applied considering only the selected symbols. To make it applicable to French narratives, we added other types of dialogue cues to the original DS markers list, which we release with the code.

Although Sini et al. (2018a) propose a rule-based algorithm focused on the French language, they relied heavily on crafted syntactic and semantic rules. Our aim was to avoid extensive manual feature engineering in order to encourage generalisation over various writing styles. Also, this method was strongly dependent on other external tools for syntactic analysis that introduced further errors too. Hence, we did not include it in the benchmark.

4.2 Deep Learning-based AADS Baselines

Deep learning-based AADS was modelled as a token classification task, which we considered more suitable for identifying *incises*. We further discuss how we preprocessed the text in order to maintain a certain degree of contextual coherence for our objective. Then, we present the two models we included in our benchmark: 1) we adapted the state-of-the-art AADS deep learning model for German (Brunner et al., 2020) to fit French language peculiarities and re-trained it from scratch , and 2) we fine-tuned CamemBERT (Martin et al., 2020) to perform sequence labelling on our dataset.

Input Preprocessing. We used spaCy (Honnibal and Johnson, 2015) to segment text in sentences and each sentence into words and punctuation.

The input length supported by contemporary language or text embedding models is limited. For instance, BERT (Devlin et al., 2019) accepts a maximum of 512 sub-word tokens, while Flair embeddings (Akbik et al., 2019) initially could handle 512 characters. This makes them unfitted to represent or produce inferences over whole books, chapters, or even larger paragraphs, which is an important limitation in computational narrative understanding. However, to preserve a certain degree of coherence within each individual text segment with regard to the DS task, we implemented an informed split as follows. Given text in reading direction, a new sentence was added to the existing segment only if the maximum input size L_C was not reached. Otherwise, the current segment was stored and a new one initialized starting with this last sentence. We discuss the choice of L_C in Section 5.

Fine-tuned CamemBERT. To specialize the general linguistic knowledge of the pre-trained language models for a precise purpose—here, to recognize DS, we use fine-tuning. We work with CamemBERT (Martin et al., 2020), one of the reference BERT-like model for French, available in the HuggingFace library (Wolf et al., 2020). However, as another tokenization of our preprocessed input is performed by CamemBERT, some adjustments were necessary to address out-of-vocabulary limitations and to handle larger sub-word sequences.

First, CamemBERT tokenizer was not be able to project all of the encoded symbols into the model's vocabulary. This was the case for breaklines as we worked with paragraphs as input, or special space encodings such as "\xa0". We spotted these unknown symbols during a first model tokenization round over the whole set of tokens, initially obtained with spacy, and replaced them with a special token *[UK]*. Another strategy could have been to remove them but we found these tokens potentially informative for AADS, as text structure cues.

Second, after the CamemBERT tokenization, a sequence of L_C tokens created during preprocessing might result in more sub-word tokens allowed as input. Similar to BERT, CamemBERT has the input limited to 512 sub-words. Here, in order to avoid the model automatically truncating the long sequence, the sequence is split in half if it overflows the input limit. Thus, it is less likely to have very short sub-sequences and context is evenly shared amongst resulting chunks. This chunking choice is closely linked to the tested L_C values (up to 512, see section Section 5). However, splits are unlikely most of the time, as SpaCy tokens are common French words—most likely represented by one or two sub-words in the model's vocabulary.

BiLTSM-CRF. We adopt the same architecture as in the state-of-the-art AADS model for German proposed by Brunner et al. (2020). Typical for sequence labelling tasks (Huang et al., 2015), it consists of two bi-directional Long-Short Term Memory (BiLSTM) layers and one Conditional Random Field (CRF) layer. The model is implemented using the SequenceTagger class of the Flair framework (Akbik et al., 2019). To embed the input, we test multiple options: Flair (Akbik et al., 2019), FastText (Athiwaratkun et al., 2018), or Flair and FastText stacked. Regarding input representation, Flair comes with a native way to handle long sequences, if these are encountered. They are chunked and each chunk is pushed to the model while keeping the last hidden state as a new hidden state[3].

5 Experiments

5.1 Evaluation Metrics

We assess the performance of models both at the token- and sequence-levels. Results are reported overall and per file. *Token-level* metrics measure the quality of the binary classification per word / token. Precision, recall and F1 scores are then computed with the scikit-learn library (Pedregosa et al., 2011). *Strict sequence match* (SSM) scores, such as precision, recall and F1 scores, measure the extent to which the predicted DS sequences strictly match the ground-truth ones. These are computed with the seqeval library (Nakayama, 2018)

We also employ another sequence-level score: *Zone Map Error* (ZME). This is our custom adaptation of the error computation method originally developed for page segmentation (Galibert et al., 2014). We include ZME because: 1) we wanted to have complementary scores that alleviate SSM's strictness; 2) we aimed to leverage it to get more insights into the quality of the output by studying the impact of various types of errors a model makes.

ZME relies on a classification of error types that depends on the overlap between ground-truth and predicted spans. The overlap can be *perfect*, *overlapping*, *including* or *included*. The error types we could obtain are then: *Match Error* (1-to-1 non-perfect overlapping between ground-truth and predicted spans), *Miss* (non-detected ground-truth DS span), *False Alarm* (falsely detected DS span), *Merge* (several ground-truth DS spans are covered by only one predicted span), or *Split* (several predicted spans within a unique ground-truth one). The score is also dependent on the span length and the number of correctly classified tokens within a span (Galibert et al., 2014). Note that this is an error score, thus it should be minimized. We present ZME in more detail in Appendix B.

A final remark is that sequences are not necessarily utterances or turns. A single turn can be split into several sequences if it contains *incises* by the narrator. Reversely, several utterances can be merged in the same sequence if they are not separated by any token labeled as non-DS (O).

5.2 Experiment Details

The deep-learning based models were trained using the train split and the best configuration was identified using the validation split. Only a part of the hyper-parameters were tuned as explained further in this section. The rule-based baselines do not need training. However, for space limitation, we report in Section 6 only the results of the best performing regex baseline on the validation split. In accordance with the task formalization and most of the existing literature, token-level F1-score was the metric used for model selection, averaged over files to mitigate the influence of longer chapters.

The two rule-based baselines exhibited similar token-level F1-scores on the validation data (over all files): 89% for BookNLP-inspired method (Bamman et al., 2014) and 87% for Byszuk et al. (2020)'s baseline. However, the BookNLP-inspired regex system showed large variance across files and scored 8 points less than its counterpart baseline adapted from (Byszuk et al., 2020) on the averaged token-level F1-score. Thus, we retained only this latter in further analyses, which we denote *Regex*.

We trained BiLSTM-CRF model for 10 epochs with a batch size of 8 and learning rate set to 0.1. After each epoch, performance was assessed on the validation set and the best configuration over epochs was retained. Regarding the input embeddings, we obtained the largest results for the stacked Flair and FastText, similar to the original work on German AADS (Brunner et al., 2020). We also benchmarked different values (from 64 to 448) for the input size L_C. Both token-level and SSM F1-scores peaked for $L_C = 192$ on the validation split, which is the value we keep for test[4].

We fine-tuned CamemBERT for 3 epochs with a batch size of 8. Similar to the experimental setup of BiLSTM-CRF, we retained the model that yielded the best results on the validation set after any epoch. We also investigated multiple input size values, L_C, from 128 to 512. For each value, training was repeated with 6 different initialisation seeds. $L_C = 320$ led to the best results. By manually analysing the sub-word sequences, we noticed that this value corresponded to the maximal input sequence length accepted by the transformer model after the inner preprocessing for length adjustment. Indeed, smaller word sequences are likely to result in sub-optimal context use while longer word sequences would more often overflow the input size accepted by the model and be automatically split.

[3] https://github.com/flairNLP/flair/pull/444

[4] The training for each L_C was done with one seed.

	Test$_C$			Test$_N$		
	Regex	*BiLSTM-CRF*	*F.CamemBERT*	*Regex*	*BiLSTM-CRF*	*F.CamemBERT*
Tok. F1	90	83	**96**	47	88	**93**
SSM F1	45	72	**76**	5.5	**30**	26
ZME	0.23	0.41	**0.09**	1.09	0.29	**0.24**
Av. Tok. F1	90 (2.3)	84 (20)	**95** (3.8)	36 (39)	82 (16)	**89** (10)
Av. SSM F1	43 (17)	71 (22)	**72** (17)	5.5 (15)	**28** (14)	24 (18)
Av. ZME	0.24 (0.05)	0.52 (0.85)	**0.11** (0.08)	1.13 (1.06)	0.55 (0.81)	**0.30** (0.23)

Table 2: Results, overall (top) and averaged over files (bottom) with standard deviations in parentheses on clean (C), and noisy (N) test-sets. Best scores are in bold.

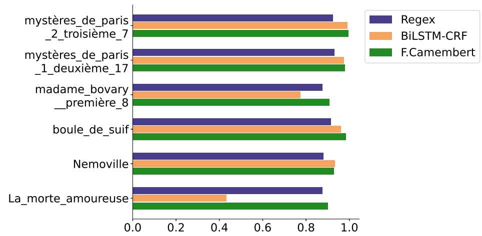
6 Results

Table 2 shows the obtained results, overall (top) and averaged over files (bottom). The scores are computed separately on clean (Test$_C$) and noisy (Test$_N$) data to assess generalization.

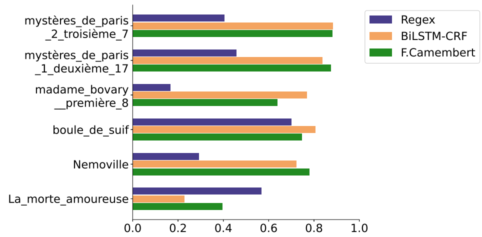
6.1 Performance on well-formatted files

The scores on Test$_C$ show that *Regex* is a strong baseline on well-formatted texts, reaching a token-level F1-score of 90% and a SSM F1-score of 45% despite its design limitations (e.g. inability to spot *incises*). The fine-tuned CamemBERT (*F.CamemBERT*) substantially outperforms *Regex* on all computed metrics, especially on span-level metrics. Though *BiLSTM-CRF* has a poorer token-level performance compared to *F.CamemBERT*, it yields a competitive SSM F1-score when averaged over files but with a larger variance. In contrast, *BiLSTM*'s ZME scores are much worse than the *F.CamemBERT*'s ones and are even worse than those of the simple *Regex*.

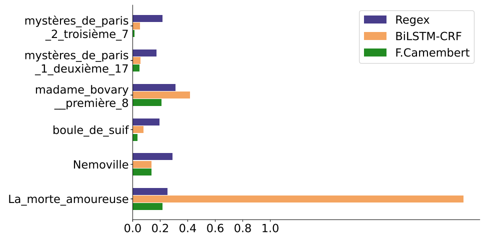
ZME depends on the span length when computing the contribution of each error type (see Appendix B) and *BiLSTM* appears to make errors concerning longer spans. Also, as further shown by the performances per file in Figure 2, *BiLSTM-CRF* struggles on *La_morte_amoureuse*. This can be, at least partly, explained by the nature of this text. The chapter from Théophile Gautier's work is homodiegetic: it is written at the first person ("je") and the character / narrator directly addresses the reader (frequent use of the second person pronoun "vous"). Thus, it could be particularly hard to distinguish DS from narration on this type of text, especially if the model indirectly relies on such cues. The *F.CamemBERT* seems more robust even in these challenging settings, although it struggles with identifying full spans in this case.



(a) Token-level F1-scores



(b) SSM F1-scores



(c) ZME score

Figure 2: Results per file from Test$_C$.

6.2 Performance on noisy files

The results on Test$_N$ allows us to get insights on the generalization capabilities of the baselines, in particular when handling low formatting quality. *Regex* displays poor generalization which was expected given its design and reliance on typographic and formatting cues. Its token-level F1-score is 53 points less compared to the clean setup in Table 2. In fact, *Regex* cannot even detect any DS token on some files as shown in Appendix C.

In contrast, deep learning-based models are less impacted by changes in formatting quality in terms of token-level F1 scores. In this regard, the *F.CamemBERT* remains the best model overall and averaged over files. *BiLSTM-CRF* shows a better overall token-level F1 score on Test$_N$ than on Test$_C$ (88% vs. 83%). As shown in Appendices A and C, it is linked to the model obtaining very good scores on chapters with many DS tokens.

Moreover, the deep learning models are much better than *Regex* on the span-level metrics. *BiLSTM-CRF* is slightly more competitive than *F.CamemBERT*, but the average over files SSM F1-scores are not significantly different. Indeed, as emphasized by the results per file in Appendix C, the performance is chapter-dependent. While *F.CamemBERT* consistently outperforms the other baselines on token-level F1-score on all files, *BiLSTM-CRF* is better at recognizing DS spans in about 22 out of 37 files (i.e. 60% of the time). However, we could notice again that the *BiLSTM-CRF*'s ZME scores are quite large but more stable than *F.CamemBERT* when the test set moves from clean (C) to noisy (N) (0.02 vs. 0.19 between the two setups). In spite of that, *F.CamemBERT* clearly appears as the best-performing model in both cases.

6.3 Qualitative analysis and Discussion

We conducted a qualitative analysis by checking the detailed contribution of each ZME error type for all models and by manually comparing a selection of files[5] with their corresponding predictions. Table 3 reveals interesting differences (despite a lack of statistical significance) between *BiLSTM-CRF* and *F.CamemBERT* on Test$_N$. While *BiLSTM-CRF* exhibits more *Miss*, *False Alarm* and *Merge* error type contributions to ZME, *F.CamemBERT*'s ZME score is more impacted by *Split* errors. The manual investigation of the selected files showed that both deep learning-based models identified much better *incises* than *Regex*. This is also consistent with the much lower *Merge* (21.8 and 12.3 vs. 633.4). Nonetheless, other semantic, syntactic or lexical cues seemed to mislead these models.

On the one hand, *BiLSTM-CRF* seemed to systematically classify parts of text written at the first person ("je") as DS, which makes it especially unfitted for homodiegetic novels (hence the low performance on *La_ morte_ amoureuse*). The punctuation seemed to be a strong cue for the model as it tended to classify sentences with exclamation or interrogation marks as DS. Then, *BiLSTM-CRF* could not handle long and uninterrupted DS paragraphs. These long DS spans often share registries or production strategies similar to narration (Egbert and Mahlberg, 2020), such as the use of past tense or descriptions, which likely misled the model.

On the other hand, the manual analysis showed that *F.CamemBERT* appeared better at identifying long DS spans or at classifying homodiegetic narration. However, this model bears other weaknesses. For instance, proper noun phrases seemed to be systematically classified as non-DS. Another common error was the miss-classification as non-DS of *[UK]* tokens in files using unrecognized non-breaking spaces (e.g. "\xa0") after quotation marks. Plus, the model regularly produced chains of alternating labels on very short groups of words as in Figure 3. These aspects correlated with the high contribution to ZME from *False Alarm* and *Split* error types.

[Une chose étonnante, c'est que je ne n']$_{DS}$éprouvais **[aucun]**$_{DS}$ étonnement d'**[une]**$_{DS}$ aventure aussi **[extraordinaire]**$_{DS}$, **[et,]**$_{DS}$ avec cette facilité que l'on a dans la vision d'admettre comme fort simples les événements les plus bizarres, **[je ne]**$_{DS}$ voyais rien là que de parfaitement naturel**[.]**$_{DS}$

Figure 3: Narration excerpt of La Morte Amoureuse by Théophile Gautier (1836) annotated by *F.CamemBERT*.

Finally, these observations also motivated a final post-processing AADS experiment. A simple heuristic is used a posteriori to hinder incoherent predictions that mix different narrative levels within the same segment of a sentence. The correction of the predicted labels post-model using a majority vote per clause lead to significant improvements on sequence-level metrics for both of the deep learning-based models. Indeed, in all settings –overall and averaged on both clean and noisy files– *F.CamemBERT*'s SSM F1 scores gained from 5 to 8 points. The performances of the *BiLSTM-CRF* model are only slightly impacted on Test$_C$ but its SSM F1 scores gained in average 5 points on Test$_N$. After this post-processing step, *F.CamemBERT* shows weaker performances than *BiLSTM-CRF* only on SSM F1 scores averaged over files on Test$_N$. Details of this clause-consistent post-processing step, as well as ensuing results, are reported in Appendix E. Altogether the different experiments tend to show that *F.CamemBERT* is the most promising model for AADS, when computational resources and ground-truth annotation are available for training.

[5]Files from Test$_C$: *La_morte_amoureuse* and *mystères_de_paris_2_troisième_7*. Files from Test$_N$: *rd0002_1*, *rd0724_0*, and *rd0367_1*.

Error Type	**Miss**	**False Alarm**	**Split**	**Merge**
Regex	1139.8 (1153.8)	63.2 (117.7)	208.0 (545.8)	633.4 (2318.0)
BiLSTM-CRF	55.6 (93.2)	144.9 (212.4)	230.1 (369.5)	21.8 (35.1)
F.CamemBERT	1.7 (4.1)	106.7 (140.0)	342.0 (566.6)	12.3 (63.6)

Table 3: Contribution of various error types to the ZME score, averaged across file, on Test$_N$. Standard deviation is reported in parentheses.

7 Limitations

The current framework bears several limitations.

First, although a common strategy in the related literature (Brunner et al., 2020; Ek and Wirén, 2019; Jannidis et al., 2018) which we also adopted, the binary annotation at the token-level is limiting. With this schema, the focus is not on speakers' utterances or turns, but on DS sequences. A subsequent issue is that consecutive turns by different characters are considered as one DS sequence if there is no "O" labeled tokens between them. One solution could have been to mark the start and end of a DS turn while paying attention to handle imbricated narration (ie. *incises*). However, this would have required significant more re-annotation efforts, which we left for a future research cycle within the proposed framework.

Second, because of copyright issues the corpus contains excerpts exclusively from a specific period, 1830-1937. Thus, the models were trained and tested on a specific type of literature and may not generalize well to other forms of narratives, in particular modern and contemporary. In this direction, the curation of the test corpus could benefit from more literary insights considering that the evaluation showed high variance of the performance over chapters. This could help to better determine the application scope of the models, and which kind of narratives require further work.

With regard to the deep neural network baselines, we did not perform an extensive parameter search and model optimisation. This could have further improved the results. However, performances on recognizing full DS spans were clearly lower than token-level metrics, which had most likely other causes. Regarding the evaluation, although we adopted ZME scores from page segmentation to have more qualitative insights, there are still other aspects we have not quantified and could be particularly relevant. For instance, does the model tend to miss the beginning, the end or some other specific parts of a DS sequence? We tried to capture some of these phenomena through our manual analysis, but it is challenging to apply it at scale without introducing methods to automatically compute metrics.

8 Conclusion

We have presented an unified framework to design and evaluate AADS in written French narratives. To our knowledge, this is the largest AADS study to date in French. We consolidated a large dataset annotated per word. Then, we benchmarked two families of baselines, rule and deep learning-based, using as inspiration AADS advances in other languages (German and English). We designed an evaluation which focuses on generalization and on learning about the advantages and weaknesses of each baseline. Results show that rule-based systems work well on bounded DS conventions (quotation marks) in clean text. Other DS formats, *incises*, and poorly formatted files pose many problems. Deep learning baselines prove to be far more robust, reaching token-level F1-scores up to 95%, but with large variance across files. Yet, recognizing full spans of DS is still challenging, even when texts have good formatting quality.

While for macro analyses in literary studies, imperfect AADS may be sufficient, some use-cases require almost perfect performance when recognizing DS spans (e.g. audiobook generation from text). If a more thorough parameter optimization might help, our qualitative analysis conveys that performance gain should be instead sought by integrating domain knowledge into the models—without feature over-engineering though. Studying the models' performances after the removal of typographic cues could lead to other insights on how to increase robustness. Multilingual language models and existing AADS corpora could be also exploited for French. Another needed step would be to transition to identifying full DS turns and their corresponding speakers, with the implied manual re-annotation efforts.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019.

FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. 2018. Probabilistic FastText for multi-sense word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Melbourne, Australia. Association for Computational Linguistics.

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

Annelen Brunner. 2013. Automatic recognition of speech, thought, and writing representation in german narrative texts. *Literary and linguistic computing*, 28(4):563–575.

Annelen Brunner, Ngoc Duyen Tanja Tu, Lukas Weimer, and Fotis Jannidis. 2020. To bert or not to bert-comparing contextual embeddings in a deep learning architecture for the automatic recognition of four types of speech, thought and writing representation. In *SwissText/KONVENS*.

Joanna Byszuk, Michał Woźniak, Mike Kestemont, Albert Leśniak, Wojciech Łukasik, Artjoms Šeļa, and Maciej Eder. 2020. Detecting direct speech in multilingual collection of 19th-century novels. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 100–104, Marseille, France. European Language Resources Association (ELRA).

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

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

Dóris AC Cunha and Marc Arabyan. 2004. La ponctuation du discours direct des origines à nos jours. *L'information grammaticale*, 102(1):35–45.

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

Jesse Egbert and Michaela Mahlberg. 2020. Fiction–one register or two?: Speech and narration in novels. *Register Studies*, 2(1):72–101.

Adam Ek and Mats Wirén. 2019. Distinguishing narration and speech in prose fiction dialogues. In *DHN*.

David K Elson and Kathleen R McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Twenty-fourth AAAI conference on artificial intelligence*.

Olivier Galibert, Juliette Kahn, and Ilya Oparin. 2014. The zonemap metric for page segmentation and area classification in scanned documents. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2594–2598. IEEE.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Peter Hühn, Jan Christoph Meister, John Pier, and Wolf Schmid, editors. 2014. *Handbook of Narratology*. De Gruyter, Berlin, München, Boston.

Henry James. 2011. *The art of the novel: Critical prefaces*. University of Chicago Press.

Fotis Jannidis, Leonard Konle, Albin Zehe, Andreas Hotho, and Markus Krug. 2018. Analysing direct speech in german novels. In *DHd*.

Murathan Kurfalı and Mats Wirén. 2020. Zero-shot cross-lingual identification of direct speech using distant supervision. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 105–111, Online. International Committee on Computational Linguistics.

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

Lattice. 2022. GitHub - lattice-8094/fr-litbank: A french litbank corpus — github.com. https://github.com/lattice-8094/fr-litbank. [Accessed 04-Aug-2022].

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Grace Muzny, Mark Algee-Hewitt, and Dan Jurafsky. 2017. Dialogism in the novel: A computational model of the dialogic nature of narration and quotations. *Digital Scholarship in the Humanities*, 32(suppl_2):ii31–ii52.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Carolin Odebrecht, Lou Burnard, and Christof Schöch. 2021. European literary text collection (ELTeC). In *COST Action Distant Reading for European Literary History*.

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.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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

Christof Schöch, Daniel Schlör, Stefanie Popp, Annelen Brunner, Ulrike Henny, and José Calvo Tello. 2016. Straight talk! automatic recognition of direct speech in nineteenth-century french novels. In *DH*, pages 346–353.

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

Aghilas Sini, Elisabeth Delais-Roussarie, and Damien Lolive. 2018a. Annotation automatique des types de discours dans des livres audio en vue d'une oralisation par un système de synthèse (automatic annotation of discourse types in audio-books). In *Actes de la Conférence TALN. Volume 1 - Articles longs, articles courts de TALN*, pages 375–382, Rennes, France. ATALA.

Aghilas Sini, Damien Lolive, Gaëlle Vidal, Marie Tahon, and Élisabeth Delais-Roussarie. 2018b. SynPaFlex-corpus: An expressive French audiobooks corpus dedicated to expressive speech synthesis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Bronwen Thomas. 2012. *Fictional dialogue: Speech and conversation in the modern and postmodern novel*. U of Nebraska Press.

Ngoc Duyen Tanja Tu, Markus Krug, and Annelen Brunner. 2019. Automatic recognition of direct speech without quotation marks. a rule-based approach. In *Digital Humanities: multimedial & multimodal*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Michael Yoder, Sopan Khosla, Qinlan Shen, Aakanksha Naik, Huiming Jin, Hariharan Muralidharan, and Carolyn Rosé. 2021. FanfictionNLP: A text processing pipeline for fanfiction. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 13–23, Virtual. Association for Computational Linguistics.

File Name	Author	Year	%DS
rd0571_0	Balzac	1841	29
rd0571_1	Balzac	1841	31
rd0127_1	Sue	1842	83
rd0444_0	Sand	1845	80
rd0444_1	Sand	1845	47
rd0724_0	Dumas	1849	2
rd0724_1	Dumas	1849	16
rd0002_0	Aurevilly	1852	18
rd0002_1	Aurevilly	1852	32
rd0623_0	FevalPP	1852	20
rd0623_1	FevalPP	1852	23
rd0616_0	FevalPP	1856	29
rd0616_1	FevalPP	1856	49
rd1169_0	Ponson	1859	64
rd1169_1	Ponson	1859	33
rd1160_0	Ponson	1860	10
rd1160_1	Ponson	1860	55
rd0730_0	About	1862	34
rd0730_1	About	1862	11
rd0305_0	Aimard	1868	81
rd0305_1	Aimard	1868	44
rd1029	Gaboriau	1867	46
rd1152_0	Gaboria	1873	52
rd1152_1	Gaboria	1873	45
rd0061_1	Zola	1873	15
rd0061_2	Zola	1873	23
rd0014_0	Verne	1877	3
rd0014_1	Verne	1877	53
rd0367_0	Gouraud	1882	23
rd0367_1	Gouraud	1882	30
rd0407_0	Malot	1878	55
rd0407_1	Malot	1878	36
rd0656_0	Ohnet	1885	46
rd0656_1	Ohnet	1885	31
rd0423_0	Mary	1886	55
rd0423_1	Mary	1886	22
rd1009	Boisgobey	1888	67

Table 4: *ST!* corpus details used as noisy test-set.

A Corpus Details

Corpus details (file names, authors, publication years and DS percentages per excerpt) are given in Table 5 for the clean (C) corpus and in Table 4 for the noisy (N) corpus.

B ZoneMap

The ZoneMap Error metric (Galibert et al., 2014) was originally developed for page segmentation. ZoneMap offers a configurable way to compute area segmentation errors based on a typology of possible errors.

Let N_T, N_P be respectively the number of positive (here, DS) spans from the ground truth, and from the model's predictions. The corresponding sets can respectively be written as $\{s_i\}_{i=1}^{N_T}$ and $\{\tilde{s_j}\}_{j=1}^{N_P}$. The length of a span s_k (given in terms of tokens) is written as $|s_k|$. Ground truth and predicted spans are grouped according to rules described further into N groups G_k, $k = 1, ..., N$. Then, the error score attributed to the model is given by:

$$E_{\text{ZM}} = \frac{\sum_{k=1}^{N} E(G_k)}{\sum_{i=1}^{N_T} |s_i|} \tag{1}$$

$$\text{where } E(G_k) = (1-\alpha_C)E_S(G_k) + \alpha_C E_C(G_k) \tag{2}$$

with $\alpha_C \in [0, 1]$. $E(G_k)$ is a linear interpolation of the segmentation error rate E_S and the classification error rate E_C within group k. Both error types can be defined purposely to penalize the model differently depending on the group type of G_k. Groups' constructions, types and compositions are defined below.

Groups are constructed based on a *link force* between true and predicted spans computed as:

$$f_{i,j} := f(s_i, \tilde{s_j}) = \left(\frac{|s_i \cap \tilde{s_j}|}{|s_i|}\right)^2 + \left(\frac{|s_i \cap \tilde{s_j}|}{|\tilde{s_j}|}\right)^2$$
$$i \in \{1, \cdots, N_T\},\ j \in \{1, \cdots, N_P\} \tag{3}$$

Non-zero links are then sorted in descending order and areas are combined into groups incrementally according to one rule: if adding a new area to a group leads to the situation where a group contains multiple ground truth or predicted areas, then such an area is not added to the group in question. This process ultimately results in five types of groups:

1. *Match*: one ground truth area overlaps with one predicted area and none of them overlap with other predicted or ground truth areas (even if the covered areas are not aligned).

2. *Miss*: one ground truth area is not covered at all by any predicted area.

3. *False Alarm*: one predicted area is not covered at all by any ground truth area.

4. *Split*: one ground truth area is covered by at least two predicted areas.

5. *Merge*: one predicted area is covered by at least two ground truth areas.

Considering the nature of the AADS task as a binary classification, spans will be used instead of areas and classification error rates will be omitted further (set $\alpha_C = 0$).

For both *Miss* and *False Alarm* groups, the segmentation error rate is strictly proportional to their length: if $G_k = \{s_i\}$, respectively $G_k = \{\tilde{s_j}\}$; the group contribution to the Zone Map error is

Split	Source	File Name	Author	Year	%DS
TRAIN	*fr-LB*	Sarrasine	Honoré de Balzac	1830	22
		Pauline	George Sand	1841	28
		Madame_de_Hautefort	V. Cousin	1856	12
		Le_capitaine_Fracasse	Théophile Gautier	1863	5
		Le_ventre_de_Paris	Émile Zola	1873	15
		Bouvard_et_Pecuchet	Gustave Flaubert	1881	4
		Mademoiselle_Fifi_nouveaux_contes-1	Guy de Maupassant	1883	17
		Mademoiselle_Fifi_nouveaux_contes-3	Guy de Maupassant	1883	19
		Rosalie_de_Constant_sa_famille_et_ses_amis	Lucie Achard	1901	68
		elisabeth_Seton	Laure Conan	1903	24
		Jean-Christophe-1	Romain Rolland	1912	10
		Jean-Christophe-2	Romain Rolland	1912	3
		Douce_Lumiere	Marguerite Audoux	1927	11
		De_la_ville_au_moulin	Marguerite Audoux	1937	9
	SPF	mystères_de_paris_1_première_1	Eugène Sue	1843	36
		mystères_de_paris_1_première_3			86
		mystères_de_paris_1_première_4			73
		mystères_de_paris_1_première_11			79
		mystères_de_paris_1_première_15			64
		mystères_de_paris_1_deuxième_1			65
		mystères_de_paris_1_deuxième_7			95
		mystères_de_paris_2_troisième_3			71
		mystères_de_paris_2_troisième_6			83
		mystères_de_paris_2_troisième_20			50
		mystères_de_paris_2_troisième_16			92
		mystères_de_paris_2_quatrième_4			86
		mystères_de_paris_2_quatrième_12			77
		mystères_de_paris_2_quatrième_15			59
		mystères_de_paris_2_quatrième_19			46
		madame_bovary__première_1	Gustave Flaubert	1857	3
		madame_bovary__première_3			19
		madame_bovary__première_9			0
		madame_bovary__deuxième_1			27
		madame_bovary__deuxième_9			24
		madame_bovary__troisième_10			13
	SB	contesse_menager	Leblanc	1924	49
		cousinebette_cecile	Balzac	1847	41
VALIDATION	*fr-LB*	export	Guy de Maupassant	1898	11
		Le_diable_au_corps	Raymond Radiguet	1923	6
	SPF	mystères_de_paris_1_première_17	Eugène Sue	1843	48
		mystères_de_paris_1_deuxième_20	Gustave Flaubert	1857	52
		mystères_de_paris_2_troisième_11			54
	SB	notaire_dedier	About	1862	39
TEST	*fr-LB*	La_morte_amoureuse	Théophile Gautier	1836	14
		Nemoville	Adèle Bourgeois	1917	32
	SPF	mystères_de_paris_1_deuxième_17	Eugène Sue	1843	51
		mystères_de_paris_2_troisième_7			60
		madame_bovary__première_8	Gustave Flaubert	1857	6
	SB	boule_de_suif	Maupassant	1870	15

Table 5: Well-formatted clean corpus details shown per file and splits (train, validation, test).

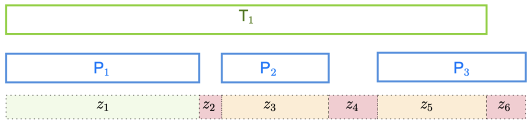


Figure 4: Split configuration type. Ground Truth span T_1 is grouped together with Predicted spans P_1, P_2 and P_3. Sub-zones z_2, z_4 and z_6 contribute fully to the error as *Miss* and *False Alarm*, while z_3 and z_5 have a mitigated contribution (factored by $(2\alpha_{MS})/3$), and z_1 doesn't contribute to the error.

$E(G_k) = |s_i|$, respectively $E(G_k) = |\tilde{s_j}|$. For *Match* groups, the group error is proportional to the number of non-overlapping tokens: if $G_k = \{s_i, \tilde{s_j}\}$, then $E(G_k) = |s_i \cup \tilde{s_j}| - |s_i \cap \tilde{s_j}|$, so that $E(G_k) = 0$ for strict span matches.

Finally, *Split* and *Merge* groups are divided into sub-zones that are in turn classified as strict *Match* and *Miss* or *False Alarm*. *Miss* and *False Alarm* sub-zones contribute to the error like *Miss* or *False Alarm* groups (strictly proportionally to the length of the sub-zones). In contrast, the largest *Match* sub-zone is not counted as an error and does not contribute to E_{ZM}, while the remaining *Match* sub-zones are partially counted as errors. Their contribution to the segmentation error rate is proportional to their length, an introduced *Merge/Split* mitigating parameter $\alpha_{MS} \in [0, 1]$ and the relative number of split, respectively merged sub-zones. Given a *Split* group $G_k = \{s_i, \{s_{j,l}\}_{l=1}^{n_k}\}$, the group is sub-divided into n_k strict match sub-zones $\{z_m\}_{m=1}^{n_k}$ and $\overline{n_k} \in \{n_k - 1, n_k, n_k + 1\}$ non-overlapping spans $\{\overline{z_m}\}_{m=1}^{\overline{n_k}}$. The segmentation error rate of such group would then be the sum of non-detected tokens $E_F(G_k) = \sum_{m=1}^{\overline{n_k}} |\overline{z_m}|$ and split penalization $E_T = \alpha_{MS} \left(\sum_{m=2}^{n_k} |z_m|\right) \frac{n_k - 1}{n_k}$. Those formula can then be rewritten in terms of original spans as:

$$E_F(G_k) = \left|s_i \cup \left(\cup_{l=1}^{n_k} \tilde{s_{j,l}}\right)\right| - \left|s_i \cap \left(\cup_{l=1}^{n_k} \tilde{s_{j,l}}\right)\right| \tag{4}$$

$$E_T(G_k) = \alpha_{MS} V_{i,j} \frac{n_k - 1}{n_k} \tag{5}$$

$$V_{i,j} = \left|s_i \cap \left(\cup_{l=1}^{n_k} \tilde{s_{j,l}}\right)\right| - \max_{l \in \{1, \cdots, n_k\}} |s_i \cap \tilde{s_{j,l}}| \tag{6}$$

and $E(G_k) = E_F(G_k) + E_T(G_k)$. *Merge* groups error contribution is computed comparably.

C Out-of-distribution results per file

Figure 5 shows the results obtained by the considered baselines on each file from Test_N, the out-of-distribution corpus.

D Computing information

We trained the models on a 32-core Intel Xeon Gold 1051 6134 CPU @ 3.20GHz CPU with 128GB RAM 1052 equipped with 4 GTX 1080 GPUs with 11GB RAM 1053 each. The required time for train, where applicable, validation and test both on Test_C and Test_N was a bit less than 2 hours: 15 minutes for *Regex*, 45 minutes for *BiLSTM-CRF*, and 40 minutes for *CamemBERT*.

E Clause-consistent predictions

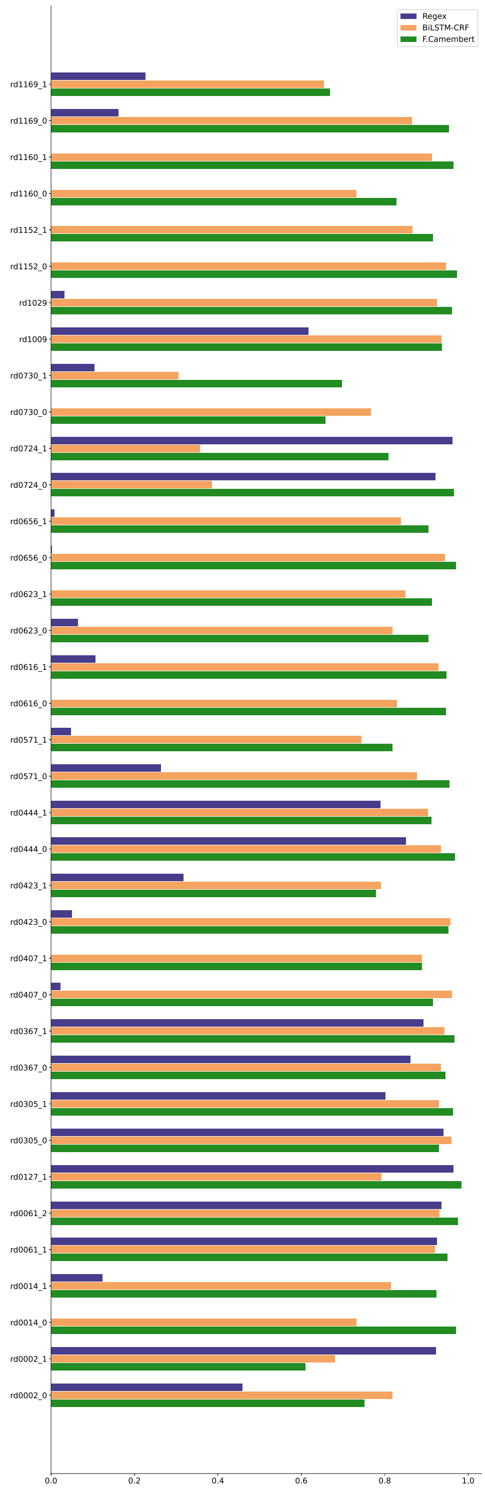
We lead a final post-processing experiment on top of the predictions made by the different models.

This step is meant to ensure the consistency of the automatic annotations at the clause[6] level. It relies on a simple heuristic drawn from the knowledge of the task: all words between two consecutive punctuation marks (full stop, question mark, hyphen, quotation mark, etc.) lie at the same narrative level, ie. the sequence of words is either uttered by a character or part of the narrator's discourse. Thus, all words stemming from a common clause must be associated with the same label.

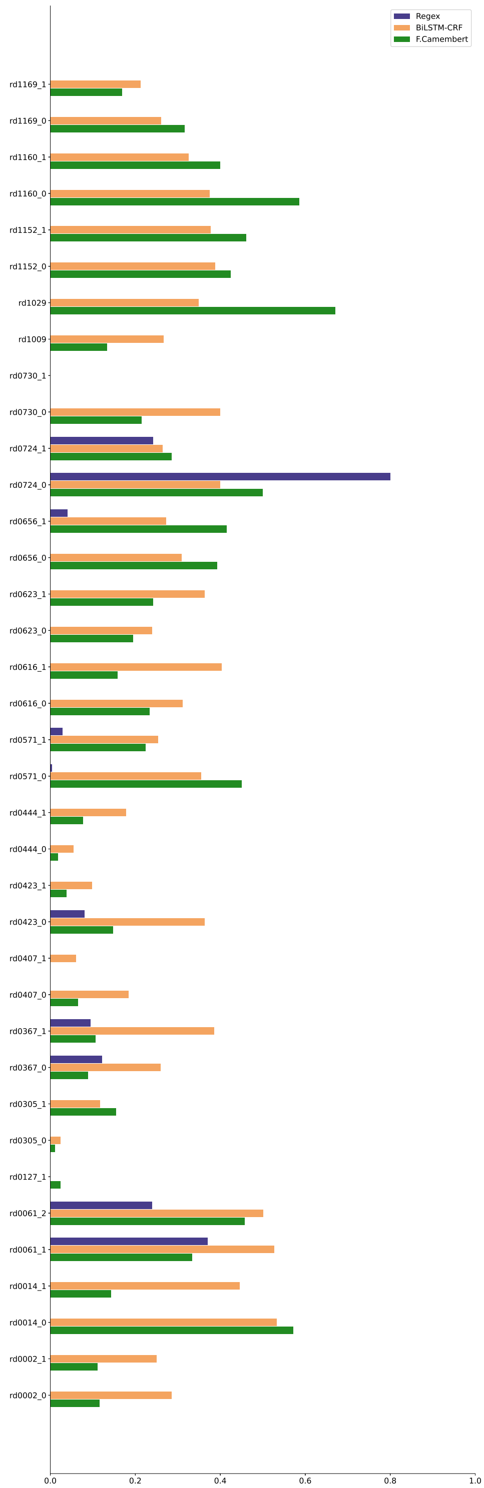
In practice, this is implemented as a post-processing step. Based on a model's predictions, the clause-consistency is ensured by imposing all words from the same clause to be associated with the same label. For each clause, a majority vote is carried out from the predicted labels to determine a consistent unique label for all the words of the clause.

Results of the clause-consistent (CC) post-processing experiment are disclosed in Table 6. As expected, this post-processing step has only few to no impact on the *Regex* model's output. Indeed, this method directly labels sequences of words caught with regular expressions that are redundant with the definition of clauses. However, imposing clause-consistent predictions allows to significantly improve the performances of the *BiLSTM-CRF* and *F.CamemBERT* based models. Overall, this heuristic never deteriorates the performances of the models on all performance scores,

[6] Here, the term *clause* does not strictly follow its grammatical definition, but it is used to designate any sequence of words between two punctuation marks.



(a) Token-level F1



(b) SSM F1

Figure 5: Results per file from Test$_N$.

	Test$_C$			Test$_N$		
	Regex	*BiLSTM-CRF*	*F.CamemBERT*	*Regex*	*BiLSTM-CRF*	*F.CamemBERT*
Tok. F1	90 =	83 =	**96** =	47 =	88 =	**93** =
SSM F1	45 =	73 +1	**81** +5	5.5 =	**34** +4	**34** +8
ZME	0.23 =	0.41 =	**0.09** =	1.09 =	0.29 =	**0.21** -0.03
Av. Tok. F1	90 (2.3) =	84 (20) =	**95** (3.8) =	36 (39) =	82 (16) =	**90** (10) +1
Av. SSM F1	43 (17) =	72 (21) +1	**78** (13) +6	5.5 (15) =	**33** (15) +5	31 (22) +7
Av. ZME	0.24 (0.05) =	0.52 (0.85) =	**0.11** (0.08) =	1.13 (1.06) =	0.54 (0.79) -0.01	**0.28** (0.23) -0.02

Table 6: Results after clause-consistent (CC) post processing, overall (top) and averaged over files (bottom) with standard deviations in parentheses on clean (C), and noisy (N) test-sets. Best scores are in bold. These scores can be put into perspectives with regards to performances of the models without CC post-processing, as displayed in Table 2. For each score, the second row shows the absolute difference between the scores obtained from the post-processed and raw predictions.

and the improvements are particularly significant for sequence-level metrics.

Enhancements are striking for the *F.CamemBERT* in all evaluation configurations. This post-processing step allows to alleviate one of the observed weaknesses of the model (see subsection 6.3) by hindering sequences of alternated labels within the same clause. This results in major performance boosts of up to 8 points for the overall SSM F1 on Test$_N$. On the other hand, sequence level performances of *BiLSTM-CRF* also benefit from the CC-predictions but mainly on noisy files with a gain of 4 points on the overall SSM F1.

Clause consistent predictions allow to reach fairly high scores even on the most challenging task of strict sequence match on well-formated documents, *F.CamemBERT* reaching on average a SSM F1 score of 78 on Test$_C$. Yet, performances on noisy files remain curtailed *F.CamemBERT* and *BiLSTM-CRF* SSM F1 scores on Test$_N$ are on average, respectively, 31 and 33 with large variances among files.

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.