The Overfitters at Touché: Joint Causality Extraction with DeBERTa-v3

Touché at CLEF 2026

Lasse Ruttert[1,*], Marvin Guelhan[1]

[1]University of Bonn, Bonn, Germany

Abstract

We describe our submission to the CLEF 2026 Touché Causality Extraction shared task, which involves detecting whether a news sentence contains causal information, extracting the text spans that participate in the causal relationship, and classifying the relationship between a given pair of spans as causal, countercausal, or uncausal. Rather than training three separate models, we train a single DeBERTa-v3-base encoder with three task-specific heads, updating all tasks jointly in a round-robin fashion every epoch. The approach rests on the observation that the sub-tasks are structurally dependent: a model that has learned to detect causal sentences should also be better positioned to extract causal spans and to classify the relationship between them. On the development set our system achieves F1 $= 0.878$ (detection), span-F1 $= 0.598$ (extraction), and macro-F1 $= 0.921$ (identification)—a $+21$ point gain on relationship identification over a single-task RoBERTa baseline. On the official test set, detection transfers well (F1 $= 0.884$), extraction reaches a granularity-F1 of 0.587, and identification drops to F1 $= 0.778$.

Keywords

Causality extraction, Multi-task learning, DeBERTa, Span extraction, CLEF 2026 Touché

1. The Problem

Consider these two sentences from the dataset, with the candidate causal spans set in italics and the pivotal connective in bold:

(a) On August 16, 2012, *police shot dead 34 people, almost all striking mineworkers*, **while** *trying to disperse and disarm them.*

(b) The policemen wrote slogans on RTC buses **without** *demanding that the transfer of Umesh Chandra be revoked.*

Both sentences mention a potentially causal event pair. In the first, the dispersal attempt *caused* the shooting: a direct causal relationship. In the second, the slogans were written *without* the demand being made. The word "without" signals an explicit *absence* of causation, a countercausal statement. A model that cannot distinguish these two produces useless output for downstream applications in knowledge graph construction or automated reasoning.

The CLEF 2026 Touché Causality Extraction task [1, 2] formalises this problem as a three-stage pipeline: **ST1** detection (is there any causal content?), **ST2** extraction (which spans are the causal participants?), and **ST3** identification (is the relationship causal, countercausal, or neither?). The dataset, drawn from newswire text, is notable for its explicit countercausal examples: sentences where causal language is present but the relationship is denied.

What makes this interesting from a modelling perspective is that the three tasks are not independent: ST1 is a prerequisite for ST2, which is a prerequisite for ST3. A model that genuinely understands causality detection should have an easier time extracting causal spans, and a model that can find those spans should in turn be better suited to classify the relationship between them. That coupling is the central motivation for our approach.

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

*Corresponding author.

s57lrutt@uni-bonn.de (L. Ruttert); s54mguel@uni-bonn.de (M. Guelhan)

2. Data: Three Tasks, One Sentence

The *countercausal_news* corpus provides 3,075 training and 340 development sentences for ST1 and ST2, and 4,603/502 for ST3. All three tasks share the same underlying sentences; the train/development partition is sentence-disjoint, so no sentence appears in both splits.

A single sentence can illustrate all three sub-tasks. Consider the example *"We have searched several jungle areas in Narayanpur and its neighbouring Kanker district in a bid to trace out the abducted cops and civilian"*, run through all three sub-tasks:

ST1 — Does it contain causal information?
Yes (label 1).

ST2 — Which spans are the causal participants?
Two character spans: *"We have searched… Kanker district"* and *"to trace out the abducted cops and civilian"*.

ST3 — How do the spans relate?
$[\text{We have searched} \ldots]_{e_0}$ in a bid $[\text{to trace out} \ldots]_{e_1}$ $\rightarrow$ **causal** (the search is motivated by the goal of finding the abducted).

Span labels in ST2 are character-level offsets into the original string. For ST3, the entity pair is marked directly in the input text with special tokens `<e0>`/`</e0>` and `<e1>`/`</e1>`, and the label distinguishes three classes: *uncausal* (0), *causal* (1), and *countercausal* (2). All three sub-tasks are evaluated with F1; ST2 uses exact character-span matching, ST3 uses macro-average over the three classes.

The class distribution in ST3 is imbalanced: uncausal and causal examples are relatively common, while countercausal examples are rarer and require picking up on negation markers ("without", "despite", "failed to") in the context of an otherwise causal-looking event pair — a pattern that later proves harder for weaker encoders to learn (Section 4), even though it is not the hardest class in absolute terms.

3. From Three Models to One

3.1. Why Joint Training?

The obvious baseline is three independent fine-tuned models, one per sub-task. We built these first, using `roberta-base` [3] with a CLS-token classification head for ST1 and ST3, and a token-level BIO sequence labelling head for ST2. Table 1 shows the results: 0.869 binary-F1 on ST1, 0.575 span-F1 on ST2, 0.708 macro-F1 on ST3. These results are reasonable, but the ST3 score in particular leaves considerable room for improvement.

The per-task model for ST3 has no idea that the input sentence was first classified as causal by ST1, or that the entity spans were extracted because they looked like causal participants to ST2. That information is simply thrown away. Joint training lets the encoder internalize all three signals at once: the detection head tells it which sentences are causally charged, the extraction head tells it which phrases carry the causal content, and the identification head sharpens the distinction between causal and countercausal framings.

3.2. Architecture

The model is a single pre-trained encoder with three task-specific linear heads, all trained simultaneously:

- **ST1 head**: $\mathbb{R}^{768} \rightarrow \mathbb{R}^2$ on the `[CLS]` token—binary classification.
- **ST2 head**: $\mathbb{R}^{768} \rightarrow \mathbb{R}^3$ on every token—BIO sequence labelling (O / B-ENT / I-ENT).
- **ST3 head**: $\mathbb{R}^{768} \rightarrow \mathbb{R}^3$ on the `[CLS]` token—ternary classification.

Dropout is applied to the encoder output before all heads. Each head uses cross-entropy loss; the ST2 head masks special and padding tokens with `ignore_index=-100`.

Training proceeds in a round-robin over the three task loaders. Each epoch runs for $S = \max(|L_1|, |L_2|, |L_3|)$ steps. Within each step we apply one forward/backward pass and one optimizer step per task in sequence, so the shared encoder receives three gradient updates per step and sees all three tasks every epoch regardless of dataset size differences. We use AdamW [4] (LR $= 2 \times 10^{-5}$, weight decay 0.01), linear warmup over 10% of total steps, gradient clipping at 1.0, batch size 16, and 5 epochs. The best checkpoint is selected by the average of the three validation metrics.

3.3. Entity Marking

For ST3, the model must know where the entity pair (e_0, e_1) is in the input. We add four special tokens (`<e0>`, `</e0>`, `<e1>`, `</e1>`) to the tokenizer vocabulary and wrap each entity span with them before tokenisation, resizing the embedding matrix accordingly. This is simpler and more expressive than providing span indices as separate inputs: the encoder can directly attend to the boundary markers in every layer, without any position-based post-processing.

3.4. Span Labelling and Offset Alignment

Sub-task 2 requires predicting character-level spans. We project gold character spans onto token-level BIO labels and decode predictions back to character offsets. One subtle issue: DeBERTa uses a SentencePiece tokenizer, whose fast `return_offsets_mapping` path is unreliable on Unicode text. We therefore compute offsets manually. The SentencePiece word-boundary marker (the underscore-like prefix at code point `U+2581`) tells us where each token begins in the original string, letting us walk forward character by character to record exact `(start, end)` positions. A related asymmetry: ST2 inputs are *not* text-normalised (no NFKC, no whitespace collapse), because normalisation would shift the gold character offsets. ST1 and ST3, which have no offset constraints, are normalised.

4. Iterating Towards the Final Model

We arrived at the submitted system through a sequence of experiments, each addressing one failure mode of the previous one. Table 1 shows all systems in order.

Step 1: per-task RoBERTa baseline. Three independent models, one per sub-task. ST1 and ST2 are serviceable; ST3 (0.708) is the obvious weak link, driven mainly by the uncausal class (F1 $= 0.655$); countercausal, despite requiring negation-sensitivity, is actually the per-task baseline's strongest class (0.752), suggesting the model relies on surface negation cues it can pick up even without joint training, while distinguishing "nothing causal is happening" from "something causal is happening but the direction is ambiguous" (uncausal) is the harder discrimination.

Step 2: joint RoBERTa. Moving to joint training with the same `roberta-base` encoder brings ST2 up by 3.3 points ($0.575 \rightarrow 0.608$) and ST3 by 4.7 points ($0.708 \rightarrow 0.755$). The shared encoder's exposure to ST1's causality signal appears to help both downstream tasks. The per-class ST3 numbers improve across the board, with uncausal seeing the largest gain ($+9.1$ pp). ST1 itself barely changes, as it was most likely already close to its ceiling.

Step 3: DeBERTa-v3-base. Swapping the encoder from `roberta-base` to `microsoft/deberta-v3-base` [5] with the same architecture and same training procedure, produces an outsized effect on ST3. Macro-F1 jumps from 0.755 to **0.921**, a $+16.6$ point improvement from the encoder change alone, and $+21.3$ points over the per-task baseline. The per-class gains are large and uniform: uncausal rises to 0.937, causal to 0.892, and countercausal to 0.932. ST1 and ST2

remain roughly unchanged (± 1 pp), suggesting that the encoder upgrade predominantly helps the relationship classification task.

A plausible explanation for this ST3-specific effect lies in DeBERTa's disentangled attention [6], which encodes content and relative position in separate matrices within each attention head. For ST3, the model must track the positional relationship between the entity markers (<e0>/<e1>) and the causal connectives ("because", "without", "despite", "in a bid to") elsewhere in the sentence: a task that should benefit directly from richer relative position representations. We present this as the most plausible explanation rather than a demonstrated one: DeBERTa-v3 also differs from RoBERTa in its ELECTRA-style pre-training and gradient-disentangled embedding sharing [5], and we have not run the ablations (e.g. disabling the disentangled-attention term, or probing marker-to-connective attention) that would be needed to isolate the attention mechanism as the cause.

What we also tried and abandoned. We experimented with replacing the ST2 BIO head with a span-pointer approach (QA-style start and end logits per position, in the style of extractive question answering), motivated by the observation that BIO tagging cannot represent overlapping spans and suffers from alignment errors at token boundaries. The span-pointer model scored 0.23–0.38 span-F1 across the hyperparameter settings we tried, consistently worse than BIO. We were unable to get it to learn to distinguish causal from non-causal spans well enough to be competitive; whether this reflects a fundamental limitation or simply an under-tuned setup we cannot say from these runs alone. The BIO ceiling at around 0.60 turns out to be the limiting factor for ST2 in this task and the one we have not overcome.

5. Results

5.1. Development Set

Table 1
Development-set results across all systems. Except for the ensemble row (Section 4), each system reflects a single training run. ST1: binary F1; ST2: span F1; ST3: macro F1.

System	ST1 F1	ST2 Span F1	ST3 F1
RoBERTa-base (per-task)	0.869	0.575	0.708
RoBERTa-base (joint)	0.883	0.608	0.755
+ ensemble (joint seeds)	0.876	0.599	0.711
DeBERTa-v3-base (joint)	0.878	0.598	**0.921**

Table 2
ST3 per-class F1 on the development set (single training run each).

System	Uncausal	Causal	Countercausal
RoBERTa-base (per-task)	0.655	0.717	0.752
RoBERTa-base (joint)	0.747	0.741	0.777
DeBERTa-v3-base (joint)	**0.937**	**0.892**	**0.932**

Training shows a clean convergence story: combined validation score improves monotonically from 0.682 at epoch 1 to 0.799 at epoch 5, with the best checkpoint selected at the final epoch. Task losses drop from $(0.46, 0.61, 0.97)$ at epoch 1 to $(0.004, 0.070, 0.183)$ at epoch 5 for ST1, ST2, and ST3 respectively. ST3 starts with the highest training loss and shows the largest absolute drop, consistent with it being the task that benefits most from the shared causality representation.

5.2. Official Test Set

We submitted the joint DeBERTa-v3-base system via TIRA [7], one run per sub-task. Table 3 shows the official test-set results for our system (*overfitters*), the other participating team (*shrome*), and the organisers' baseline, as provided by the task organisers. ST2 was officially evaluated only on the causal samples of the test set and scored with a granularity-based F1; the ST2 values therefore differ slightly from the TIRA leaderboard, while ST1 and ST3 are identical to it.

Table 3
Official test-set results for the participating systems and the organisers' baseline. ST2 is evaluated on the causal samples only and reports granularity-based F1; ST1 and ST3 report F1. Our system is *overfitters*.

Sub-task	System	F1	Precision	Recall
ST1 Detection	**overfitters (ours)**	**0.884**	0.938	0.836
	shrome	0.869	0.848	0.890
	baseline	0.872	0.872	0.872
ST2 Extraction	overfitters (ours)	0.587	0.731	0.584
	shrome	**0.728**	0.764	0.787
	baseline	0.665	0.749	0.709
ST3 Identification	overfitters (ours)	0.778	0.751	0.818
	shrome	0.817	0.796	0.851
	baseline	**0.839**	0.827	0.853

Detection generalises well: our test F1 of 0.884 slightly exceeds the development-set score (0.878) and is the best among the submitted systems, with markedly high precision (0.938) at the cost of recall.

Extraction is our weakest sub-task relative to the field: 0.587 granularity-F1, below both the organisers' baseline (0.665) and shrome (0.728), with recall (0.584) again the limiting factor. A direct development-to-test comparison is not possible here, because the development-set figure of 0.598 is an exact character-span F1 whereas the official test metric is granularity-based and computed on causal samples only. At face value, however, the test score does not indicate a dramatic degradation; rather, it confirms that our BIO approach saturates around 0.6 (Section 4) while other approaches demonstrably reach higher.

Identification drops from 0.921 on the development set to 0.778 on the test set, placing us below both shrome (0.817) and the baseline (0.839). We see two non-exclusive explanations for this gap. First, the development set was used both for checkpoint selection and for iterating over architectures and encoders (Section 4), so the reported development-set score is optimistically biased; the striking 0.921 in particular should be read as partially fitted to the development data. Second, the test set may differ from the development set in the distribution of sentence sources or class balance. With a single submitted run per sub-task we cannot decompose the gap further.

6. What We Learned

The results point to a few conclusions, which we state with the caveat that they rest largely on single-run numbers and would need multi-seed confirmation to be put on firm statistical footing.

The hierarchical structure of the tasks plausibly acts as a useful inductive bias. The ST3 improvement from per-task to joint RoBERTa (+4.7 pp), together with the parallel ST2 gain, is consistent with the encoder learning a shared representation of causal language that transfers across the three task heads. We cannot, however, cleanly separate this from a regularisation effect on the basis of one run per configuration, and the seed spread implied by the failed ensemble (Section 4) is large enough that we treat the magnitude of this gain as suggestive rather than settled.

DeBERTa's position sensitivity is a plausible driver for span-marked classification. The +16.6 pp jump from RoBERTa-joint to DeBERTa-joint on ST3, with ST1 and ST2 almost unchanged,

points to something specific to ST3's structure rather than a uniformly better encoder: ST3 is the only task where the model must relate two marked spans to each other and to the connective language between them, and disentangled position attention is well-suited to that. As noted in Section 4, the v3 pre-training changes are an alternative explanation we have not ruled out.

Development-set gains only partially survived contact with the test set. The official results (Section 5.2) confirm ST1 and are broadly consistent for ST2 (under a changed metric), but revise ST3 downwards by 14 points—from the standout result of our development experiments to below the organisers' baseline. The lesson for future participation is procedural as much as architectural: iterating over encoders and architectures against a single development split, and selecting checkpoints on that same split, inflates development-set scores in ways that a held-out internal test split or cross-validation would have exposed earlier.

Span extraction is the unsolved sub-problem. BIO tagging with either encoder plateaus at roughly 0.60 span-F1 on the development set, and the official test score (0.587 granularity-F1, the lowest among the submitted systems) confirms it as our weakest sub-task. This is a known limitation: gold spans are character-level, tokenisation introduces alignment noise, and BIO cannot represent overlapping candidates. A span enumeration approach with biaffine scoring [8] or a generative extraction model seems the natural next step.

7. Related Work

Causality extraction has a long history in NLP, spanning early shared tasks, dedicated corpus-building efforts, and more recent neural approaches. Countercausal language has received comparatively little attention, making this shared task's focus on it particularly valuable.

Multi-task learning with shared pre-trained encoders is a well-established means of improving performance on related tasks. Our setup is closest in spirit to MT-DNN [9], which shares a pre-trained encoder across NLU tasks with task-specific heads; we differ in using a round-robin per-step update rather than task-specific mini-epochs. Our observation that ensembling hurts the joint model is consistent with the broader finding that multi-task optimisation can produce sharper minima across which averaging diverse seeds is destructive.

DeBERTa-v3 [5] is a strong baseline for most NLU tasks; its particular advantage for span-marked relationship classification, as observed here, has not, to our knowledge, been specifically documented for causal extraction.

8. Conclusion

We trained a single DeBERTa-v3-base encoder jointly on all three causality extraction sub-tasks and submitted it as our system to CLEF 2026 Touché. On the development set, relationship identification (ST3) responded dramatically to both joint training and the DeBERTa encoder, gaining $+21$ percentage points over the per-task RoBERTa baseline, most plausibly driven by the encoder's ability to track the positional relationship between entity markers and causal connectives, though we have not isolated this mechanism from the other differences between the encoders. The official test results temper this picture: detection generalised well (F1 $= 0.884$, the best among the submitted systems), but identification dropped to 0.778—below both the other participating team and the organisers' baseline—indicating that part of the development-set gain reflected overfitting to the development split, which we had used for both model selection and iteration. Span extraction (ST2) remains the hardest open sub-problem: neither encoder choice nor architectural experiments pushed it past the ceiling imposed by BIO token-level labelling (0.587 granularity-F1 on the test set), and it is where our system trails the other submissions most clearly.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude (Anthropic), including Claude Code, in order to: assist with the implementation of the system and aid in writing (e.g. drafting and editing). After using these tools, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, C. Mathews, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, Overview of touché 2026 — argumentation systems, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. S. Salido, A. Barrón-Cedeño, A. G. S. Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. 17th International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[2] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, C. Mathews, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, Overview of touché 2026 — argumentation systems — extended version, in: E. S. Salido, A. Barrón-Cedeño, A. G. S. Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[4] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[5] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: 11th International Conference on Learning Representations (ICLR 2023), 2023. URL: https://openreview.net/forum?id=sE7-XhLxHA. arXiv:2111.09543.

[6] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with disentangled attention, in: 9th International Conference on Learning Representations (ICLR 2021), 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.

[7] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous integration for reproducible shared tasks with tira.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.

[8] J. Yu, B. Bohnet, M. Poesio, Named entity recognition as dependency parsing, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6470–6476. doi:10.18653/v1/2020.acl-main.577.

[9] X. Liu, P. He, W. Chen, J. Gao, Multi-task deep neural networks for natural language understanding, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 4487–4496. doi:10.18653/v1/P19-1441.