Shrome at Touché: Soft-Vote Ensembling and Counter-Causal Augmentation for Causality Extraction

Touché at CLEF 2026

Roham Zendehdel Nobari[1,*,†], Shayan Sooratgar[1,†]

[1]Department of Computational Linguistics, University of Zurich, Switzerland

Abstract

Touché 2026 extends causality extraction to counter-causal claims: news sentences whose surface form appears causal but whose meaning denies the causation, as in *"It is falsely believed that X caused Y."* A system that relies on surface cues such as *caused* or *led to* will accept such a sentence as causal and give it the wrong polarity. On the Countercausal News Corpus (CCNC) [1], the task has three subtasks: deciding whether a sentence is causal (detection), locating its cause and effect spans (extraction), and labeling its polarity as procausal, counter-causal, or uncausal. We build one model per subtask. Detection is a fine-tuned classifier with a single cross-task rule that uses the extracted spans to remove false positives. For extraction, we ensemble three RoBERTa-large [2] BILOU+CRF taggers by averaging their token-level scores before decoding, rather than voting on the spans each tagger produces. For polarity, where labeled counter-causal examples are scarcest, we add training sentences generated by a large language model prompted with nine patterns of counter-causal expression adapted from Hagen et al. [1], keeping only those that pass automatic structural checks. On the held-out CCNC test set, the system reaches F_1 0.869 on detection and macro-F_1 0.817 on polarity, and in the organizers' final causal-only evaluation of extraction it scores granularity-adjusted F_1 0.728, the highest extraction score among all submissions including the organizers' baseline. The development split is used only for component selection and the ablations reported in the paper.

Keywords

Causality extraction, Counter-causal classification, Cause-effect span extraction, Emission-level CRF ensembling, LLM-based data augmentation, Causal News Corpus

1. Introduction

Causal claims pervade news text, with reports routinely asserting that one event brought about another. Automatically extracting these claims supports downstream causal reasoning. The difficulty is that many causal statements in public discourse are disputed, outdated, or explicitly denied. A claim such as *"sugar causes hyperactivity"* is widely repeated yet contradicted by the evidence, and a news sentence may invoke a cause-effect pair precisely in order to reject it. Research on causality extraction has largely neglected such *counter-causal* statements [1], even though telling an asserted cause apart from a denied one is essential to representing causal knowledge faithfully.

The Touché 2026 lab on Argumentation Systems [3, 4] targets this gap with its Task 2, Causality Extraction, built on the Countercausal News Corpus (CCNC) [1]. Relative to prior shared tasks on the Causal News Corpus [5, 6], the central change is the presence of counter-causal sentences whose surface form contains an apparent cause-effect pair but whose discourse *denies* the causation (*"It is falsely believed that X caused Y."*, *"It is not that X caused Y."*). A model that detects causality from surface lexical cues alone – the verbs *caused*, *led to*, *resulted in* – will accept such sentences as causal and assign the wrong polarity. The task decomposes into three subtasks. **Detection** is binary classification of whether the sentence is causal, **extraction** locates the cause and effect spans, and **polarity** labels a given sentence-and-pair as procausal, counter-causal, or uncausal.

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

*Corresponding author.

†These authors contributed equally and share first authorship.

roham.zendehdelnobari@uzh.ch (R. Z. Nobari); shayan.sooratgar@uzh.ch (S. Sooratgar)

We use *procausal* for what Hagen et al. [1] call the *causal* polarity class, to avoid the overload with the detection subtask's binary *causal* label. We treat the three subtasks as separate models with one cross-task interaction in which extraction-stage spans gate the detection-stage positive output. Detection itself is a fine-tuned BERT classifier with that single gating rule, so our two methodological contributions fall in extraction and polarity; the third reports the full system. Our contributions, organized by subtask, are as follows.

1. **Extraction (emission-level soft-vote ensembling).** A single RoBERTa-large [2] model with a three-layer BILOU+CRF [7] tagger, trained on the CCNC extraction split plus the external CNCv2/RECESS corpus [8] with EDA augmentation [9], reaches 0.5742 exact-match F_1 on the CCNC development split before any ensembling; this base supplies most of our extraction performance. Our contribution is to combine three such seeds by averaging per-token CRF emissions and transition matrices before a per-layer Viterbi decode, rather than voting over already-decoded spans. On a matched three-seed panel this emission-level soft-vote beats span-level hard voting by $+1.07$ pp exact-match F_1, with a parallel gain on the more lenient overlap metric, and the production panel reaches 0.5933 (Section 6.2). Emission averaging is a standard CRF-ensembling pattern; the contribution is its matched-panel measurement on the Causal News Corpus family, where prior CASE submissions did not report this comparison.
2. **Polarity (taxonomy-conditioned counter-causal augmentation).** We generate counter-causal training data with a GPT-class large language model conditioned on nine patterns of counter-causal expression adapted from Hagen et al.'s Table 4 inventory and validate every candidate with mechanical structural checks. The structural-validator acceptance rate on the production generation pass is 93.2%, a syntactic pass rate that does not certify semantic correctness. This augmentation is the main lever for polarity. At a fixed seed, two augmentation passes raise dev macro F_1 from 0.766 (no augmentation) to 0.890, with the per-pass breakdown reported in Section 7.2.
3. **Full system (all three subtasks).** Integrating the per-subtask models into a single pipeline, the system reaches binary F_1 0.869 on detection and macro F_1 0.817 on polarity on the held-out test set, and ranks first on extraction in the organizers' final causal-only evaluation with granularity-adjusted F_1 0.728 (Section 8 reports the TIRA scores, development results, and the dev-to-test gap).

2. Related Work

Causality extraction and the Causal News Corpus. Recent work on causality extraction from news centers on the Causal News Corpus. CNCv1 [5] annotates English news sentences from Indian outlets for binary causal classification; its successor CNCv2, also released as RECESS [8], adds cause/effect/signal span annotation. The two underpinned the CASE 2022 [6] and CASE 2023 [10] Event Causality Identification shared tasks. These resources treat a sentence as either asserting a causal relation or not, with no category for sentences that deny one. Hagen et al. [1] address this gap with the Countercausal News Corpus, which extends CNCv2 by manually rewriting roughly half of its causal sentences into counter-causal form, producing a three-way procausal/counter-causal/uncausal corpus (1,028 / 952 / 1,435 sentences). They also provide RoBERTa baselines on its validation split (macro-averaged: detection F_1 0.874, BIO-tag F_1 0.440, ternary identification F_1 0.921), computed under metric and split conventions that differ from those of the shared task and so are not directly comparable to results reported here.

Span extraction with encoder–CRF taggers. Pairing a neural encoder with a linear-chain CRF [11] over the BILOU tagging scheme [12] has been the standard recipe for span extraction since the BiLSTM-CRF tagger of Lample et al. [13], later restated with transformer encoders in place of the BiLSTM. For causal spans specifically, the BoschAI submission to CASE 2023 [7] ranked first on Subtask 2 using a multi-layer BILOU+CRF tagger over a shared transformer encoder, stacking simultaneous cause-

effect-signal relations through pipe-concatenated labels $\langle t_1|t_2|t_3\rangle$ that admit up to three relations per sentence.

Data augmentation for low-resource labeling. Easy Data Augmentation (EDA) [9] enlarges training sets through synonym replacement and random word operations, but its sentence-level edits disturb token-level span boundaries; Dai and Adel [14] instead design label-aware token-level operations for sequence labeling, replacing tokens or whole mentions in a way that keeps the tag sequence consistent. A separate line synthesizes new labeled examples with large language models, e.g., Schick and Schütze [15], which is attractive for enlarging rare classes that have little annotated data.

NLI as a zero-shot classifier and verifier. Yin et al. [16] showed that off-the-shelf natural-language-inference models can act as zero-shot text classifiers by phrasing each candidate label as an entailment hypothesis, and the same paradigm has since been used to verify generated or extracted text. Strong multi-corpus checkpoints, such as the DeBERTa-v3 [17] model of Laurer et al. [18] trained on MNLI [19] and several further NLI datasets, make this usable without task-specific training.

Task-adaptive pre-training. Gururangan et al. [20] show that continuing masked-language-model pre-training on in-domain or task text before fine-tuning improves downstream accuracy, a low-cost adaptation when the labeled target corpus is small.

3. Dataset

CCNC [1] extends CNCv2 [8] by manually rewriting roughly half of CNCv2's causal sentences into counter-causal form. We train on the organizer-provided splits (Table 1); the extraction and polarity models additionally use in-domain augmentation, described in Sections 6 and 7.

Table 1
CCNC counts on the splits used. Detection and extraction share sentence-level data; polarity has more rows because each candidate cause/effect pair within a sentence is annotated separately.

Subtask	Train	Dev	Gold-span or class units
Detection (binary)	3,075	340	1 label per sentence
Extraction	3,075	340	473 gold spans on dev
Polarity (3-class)	4,603	502	2,412/1,305/886 train; 264/120/118 dev

The polarity training set is class-skewed, and the smallest class, counter-causal, carries explicit negation cues (*not*, *n't*, *never*, *falsely*, *contrary*, *denied*) far more often than the other two (Table 2). This roughly four-fold cue gap between procausal and counter-causal motivates the negation-scope feature discussed in Section 9.

Table 2
Polarity training composition. The counter-causal class is the smallest yet shows explicit negation cues most often (cue prevalence measured on dev). The nine-pattern LLM augmentation of Section 7 adds 2,433 rows, raising it from 886 to 3,319.

Class	Train rows	Neg. cues (dev)	After aug.
uncausal	2,412	29.5%	2,412
procausal	1,305	15.8%	1,305
counter-causal	886	67.8%	3,319
Total	4,603	—	7,036

4. System Overview

Three subtask models run independently on the same input sentence (Figure 1): a BERT-base binary detector, a three-seed emission-level soft-vote ensemble of three-layer BILOU+CRF taggers on RoBERTa-large for extraction, and a four-seed mean-softmax ensemble of BERT-base classifiers with PURE typed entity markers around cause and effect spans for polarity. A single-rule cross-task stacker overrides detection to *uncausal* whenever extraction produces no spans, correcting about twice as many dev false positives as the new false negatives it introduces (exact flip counts in Section 5.1), without retraining either model.

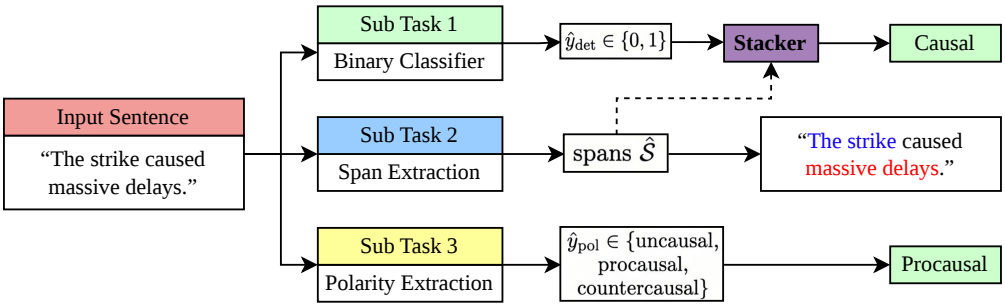

Figure 1: System overview on an illustrative input sentence. The detection, extraction, and polarity models are trained independently; the only cross-task interaction is the dashed arrow, through which the predicted spans $\hat{\mathcal{S}}$ gate the detection output via the one-rule stacker (causal iff $\hat{y}_{\text{det}} = 1$ and $|\hat{\mathcal{S}}| \geq 1$).

5. Detection

5.1. Architecture, training, and the cross-task stacker

We fine-tune `bert-base-uncased` [21] with the default sequence-classification head over the pooler output (a tanh layer on the `[CLS]` hidden state), shown in Figure 2. Inputs are raw sentences truncated at 128 WordPiece tokens.

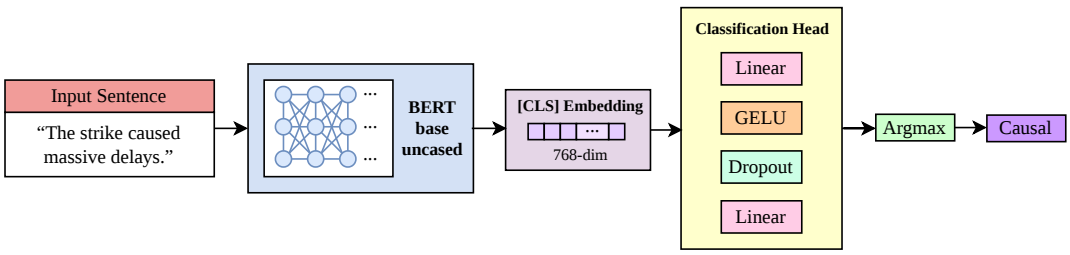

Figure 2: Detection model. The input sentence is encoded by BERT-base; the `[CLS]` representation passes through the stock sequence-classification head, and an argmax over the two logits yields the binary causal/uncausal label.

Training is a standard fine-tune at a single seed (hyperparameters in Appendix A). No early stopping is configured: the model trains a fixed five epochs and the best-validation-loss checkpoint is restored at the end. We did not invest heavily in detection: a TF-IDF plus logistic-regression baseline already reaches 0.793 binary F_1 on dev, leaving headroom that BERT-base closes (Table 3).

The cross-task stacker is a one-rule override that flips the detector's positive output to *uncausal* when the extraction stage predicts no spans: if no cause or effect span is located, no causal relation can be reported. Formally, we predict *causal* iff $\hat{y}_{\text{det}} = 1$ *and* $|\hat{S}| \geq 1$, and *uncausal* otherwise. This rule was selected by grid search over ten candidates on the dev split: one detector-only rule plus disjunctive (OR), flip, and conjunctive (AND) variants parameterized by the minimum span count. Over the standalone

detector (Table 3), the conjunctive rule with span threshold one maximizes dev causal-class F_1 at 0.8946, a 2.6 pp gain.

The rule flips 46 dev sentences from causal to uncausal: 31 flips correct existing false positives and 15 introduce new false negatives. The calibration log accordingly records causal-class recall over the 207 gold causal sentences dropping from 0.9130 standalone to 0.8406 stacked, a recall loss that the precision gain outweighs on this split. The rule is only safe while extraction recall on causal sentences stays high enough that the span-presence gate does not start removing true positives. With ten candidates evaluated on a 340-sentence dev split, the gain partly reflects selection over the rule space. We did not submit the unstacked detector to the TIRA test split, so we cannot quantify how much of the dev gain transferred; the stacked test causal-class F_1 of 0.869 (Section 8) is consistent with either partial transfer plus underlying detector drift, or full transfer on a harder underlying split.

5.2. Baselines and dev results

Table 3 summarizes the detection dev results. The simple TF-IDF baseline outperforms the BiLSTM as well as both off-the-shelf causal systems (the UniCausal sequence baseline and zero-shot BART-MNLI), fine-tuned BERT-base clears every baseline by a wide margin, and the span-gated stacker of Section 5.1 adds the final increment and is the production configuration.

Table 3
Detection dev results (binary F_1). The production system is the stacked variant in the last row. UniCausal baseline is the sequence-classification head of Tan et al. [22]; BART-MNLI is the zero-shot pipeline over `facebook/bart-large-mnli`.

Configuration	Binary F_1 (causal class)
Random	0.490
TF-IDF + logistic regression	0.793
BiLSTM	0.731
UniCausal sequence baseline	0.553
Zero-shot BART-MNLI	0.642
BERT-base (seed 42, standalone)	0.8690
Stacker: BERT-base $\wedge$ (extraction has spans)	**0.8946**

6. Extraction

6.1. Per-model architecture: three-layer BILOU+CRF

Each seed adapts the multi-layer BILOU+CRF idea of BoschAI [7] into a parallel-head form (Figure 3): three independent linear BILOU heads on a shared RoBERTa-large encoder [2], each producing five-way per-token logits over $\{O, \text{B-ENT}, \text{I-ENT}, \text{L-ENT}, \text{U-ENT}\}$ – Outside, Begin, Inside, Last, and Unit (single-token span) – for a single ENTITY type (CCNC's extraction-stage annotation does not distinguish cause from effect, so one label suffices) and decoded by its own linear-chain CRF [11]; in contrast to BoschAI's pipe-concatenated composite labels (Section 2), the three layers do not share output space or transition matrix. Gold spans within a sentence are greedily colored onto the three layers by an interval-graph algorithm; no CCNC training sentence requires more than three layers, so none are dropped. The loss averages the per-layer negative CRF log-likelihood, and for the TAPT seed adds a uniform label-smoothing auxiliary cross-entropy: a simplification of the targeted boundary-smoothing scheme of Zhu and Li [23] that retains the regularization intent without boundary-localized reweighting. Two training choices matter beyond defaults: the CRF parameters sit in a separate optimizer group at a roughly six-fold higher learning rate than the encoder, and early stopping monitors overlap F_1 rather than validation loss. All hyperparameter values are listed in Appendix A.

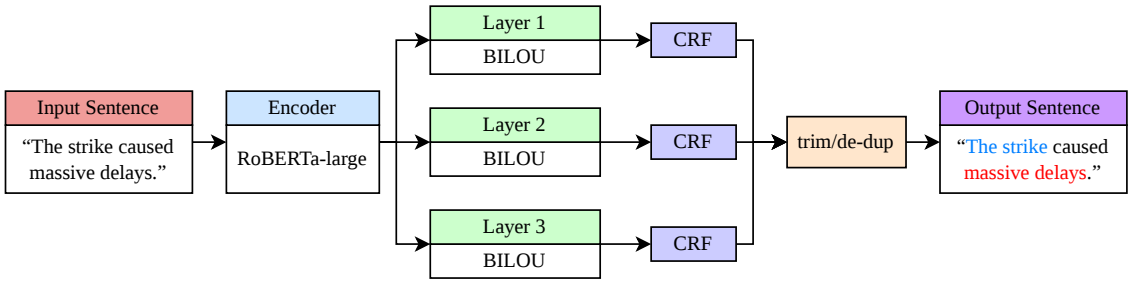

Figure 3: One extraction seed. A shared RoBERTa-large encoder feeds three parallel BILOU layers, each decoded by its own linear-chain CRF; the decoded spans are unioned across layers, de-duplicated, and trimmed to produce the output spans.

Training data. CCNC alone is small enough that the tagger overfits it: a RoBERTa-large model trained on the 3,075-row extraction split without augmentation, with validation-loss early stopping, reaches only 0.40 exact F_1 on dev (training loss collapses while recall stalls). Adding external in-domain data and EDA augmentation, and switching the early-stopping monitor from validation loss to overlap F_1, together lift the best single seed to 0.5742 (Table 5); we did not run a one-factor-at-a-time ablation, so we report these as a joint effect.

The ensemble averages three independently trained seeds, where a *seed* is one complete fine-tuning run (Section 6.2): two follow a *base recipe* and one a *TAPT recipe* (seeds listed in Appendix A). Both recipes add CNCv2 (RECESS) [8], converted to the CCNC schema with signal tokens stripped (keeping the signal annotations costs about 6 points of overlap F_1, as the model then spends capacity on a span class absent from CCNC), and apply EDA [9] (synonym replacement and random insertion; one augmented copy per sentence with fewer than three spans) plus a double oversample of sentences with three or more spans. The TAPT recipe differs in three ways: it (i) warm-starts from a *task-adaptively pre-trained* encoder [20], that is, one produced by continued masked-language modeling on the unique CCNC and CNCv2 sentences, driving the masked loss from 1.67 to 0.36 (schedule and corpus size in Appendix A), rather than off-the-shelf RoBERTa-large; (ii) concatenates the UniCausal [22] grouped rows; and (iii) adds the boundary-smoothing loss described above. Table 4 gives the resulting pool sizes.

Table 4
Extraction training-pool composition by recipe. Source rows are expanded by EDA augmentation and multi-span oversampling; the two base seeds differ slightly (seed 123: 11,489; seed 2026: 11,492 rows) and the TAPT seed reaches 13,432.

Source	Base recipe	TAPT recipe
CCNC extraction train	3,075	3,075
CNCv2/RECESS (signals stripped)	3,075	3,075
UniCausal (signals stripped)	—	1,049
Source rows	6,150	7,199
After EDA + oversampling	≈11,500	≈13,400

Decoding. At inference we collect per-layer BILOU logits, Viterbi-decode each layer through its own CRF, map tag sequences to character-level spans via the RoBERTa offset map, and union spans across the three layers, deduplicating exact matches and trimming whitespace and the 15 punctuation characters ", ’, ‘, ,, ., ;, :, !, ?, (,), [,], {, } from span endpoints.

6.2. Emission-level soft-vote across three seeds

The final ensemble averages CRF emissions and transition matrices across three seeds before a per-layer Viterbi decode through each averaged CRF, rather than voting over predicted spans (Figure 4). A

single seed of the architecture above reaches dev exact F_1 of 0.5742 and overlap F_1 of 0.7692 (seed 123); hard span-level voting across three or four seeds adds only a marginal exact-F_1 gain (Table 6). Let N denote the number of subword tokens in the sentence and identify the BILOU tag set $\{O, B, I, L, U\}$ with $\{1, \ldots, 5\}$. Given K models each producing L layers of emissions $\mathbf{E}_k^{(\ell)} \in \mathbb{R}^{N\times 5}$, transition matrices $\mathbf{T}_k^{(\ell)} \in \mathbb{R}^{5\times 5}$, and start and end vectors $\mathbf{s}_k^{(\ell)}, \mathbf{e}_k^{(\ell)} \in \mathbb{R}^5$, define the cross-seed mean $\bar{\mathbf{X}}^{(\ell)} = K^{-1}\sum_{k=1}^{K} \mathbf{X}_k^{(\ell)}$ for each tensor. With $K = L = 3$ in our system, we Viterbi-decode

$$\arg \max_{y_{1:N}\in\{1,\ldots,5\}^N} \bar{\mathbf{s}}_{y_1}^{(\ell)} + \sum_{n=1}^{N} \bar{\mathbf{E}}_{n,y_n}^{(\ell)} + \sum_{n=1}^{N-1} \bar{\mathbf{T}}_{y_n,y_{n+1}}^{(\ell)} + \bar{\mathbf{e}}_{y_N}^{(\ell)}$$

separately for each layer $\ell \in \{1, \ldots, L\}$, with span union across the L decoded sequences and the trimming of Section 6.1.

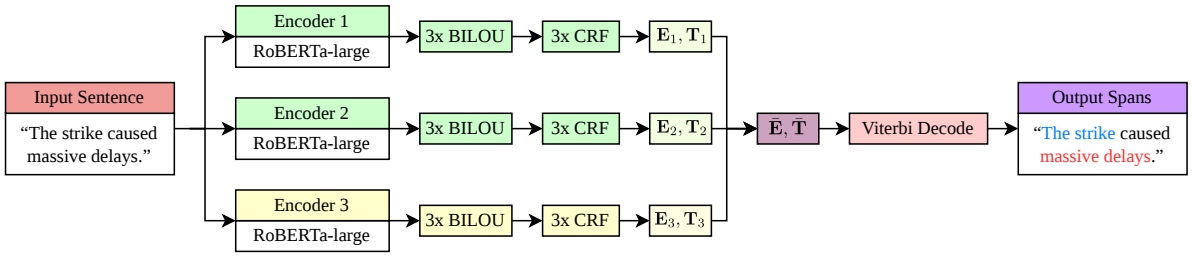

Figure 4: Emission-level soft-vote. Each of the three seeds produces per-layer emissions and transition scores $(\mathbf{E}_k, \mathbf{T}_k)$; these are averaged into $(\bar{\mathbf{E}}, \bar{\mathbf{T}})$ *before* a single per-layer Viterbi decode, so near-miss agreement between seeds survives to the decoding step instead of being discarded by span-level voting. The two green encoders are the base-recipe seeds; the yellow encoder is the seed warm-started from the task-adaptively pre-trained (TAPT) checkpoint (Section 6.1).

Emission-level voting recovers near-miss agreement that span-level voting discards: when models A, B, and C assign START probabilities of 0.80, 0.60, and 0.70 at position n, the averaged decode sees 0.70 even if no individual decoder emitted a START there. On a matched three-seed panel, swapping span-level hard-vote for emission-level soft-vote adds 1.07 pp exact and a comparable overlap gain (Table 6). Averaging CRF tensors before joint Viterbi is a standard CRF-ensembling pattern; the BoschAI submission [7] fine-tuned a single multi-layer BILOU+CRF model without ensembling, and to our knowledge emission-level soft-vote across seeds on multi-layer BILOU+CRF panels has not been reported for the Causal News Corpus family.

The production panel. After a sweep over candidate triples we select seeds 123 and 2026 on the base recipe and seed 42 on the TAPT recipe with the boundary-smoothing auxiliary loss, the triple that maximizes the mean of exact and overlap F_1 on dev (Table 5). This panel differs from the matched comparison above because one seed carries the TAPT warm-start, so its endpoint cannot be read directly from the hard-versus-soft delta. The TAPT recipe helped only seed 42 (+1.17 pp exact) and hurt the other three seeds we tested by 1.6 pp to 3.0 pp, averaging -1.38 pp across all four paired seeds. We retain the TAPT seed for emission diversity through a different encoder warm-start, not as a method-level endorsement of TAPT (Section 9).

6.3. NLI span verifier

As a post-hoc complement to the supervised pipeline, an off-the-shelf entailment model independently accepts or rejects each span produced by the soft-vote panel. For each span s we form the hypothesis `"This sentence describes a causal event involving 's'."` and feed (premise = original sentence, hypothesis) into a multi-corpus DeBERTa-v3-large NLI checkpoint [18] trained on MNLI plus four additional NLI corpora, retaining spans with entailment probability at least τ. The threshold

Table 5
Per-seed and ensemble dev F_1 for the production extraction panel.

Seed	Encoder warm-start	Recipe	Exact F_1	Overlap F_1
123	RoBERTa-large	base recipe	0.5742	0.7692
2026	RoBERTa-large	base recipe	0.5505	0.7569
42	RoBERTa-large + TAPT	+UniCausal, boundary-smoothing aux.	0.5260	0.7584
Soft-vote of the three seeds above			**0.5933**	**0.7800**

$\tau = 0.1$ maximizes exact F_1 on dev, removing only the three most clearly spurious spans; thresholds $\tau \geq 0.2$ degrade F_1 monotonically by removing borderline true spans. An earlier pair-level template "s_{cause} `caused` s_{effect}" lost 5.8 pp F_1: gold CCNC cause and effect spans do not compose into well-formed directional hypotheses. The pair *"agitating ex-servicemen accusing the government"*, for example, receives an entailment probability of only 0.25 under that template.

Applied to the soft-vote panel, the verifier produces a within-noise change on dev: it removes three predicted spans, each a truncated fragment of a gold span, trading a marginal exact-F_1 gain for a comparable overlap-F_1 loss (Table 6). We discuss this within-noise gain in Section 9.

6.4. Final results

Table 6 tracks the dev progression: emission-level soft-vote lifts the single-seed baseline by 1.91 pp exact to the production panel's 0.5933, and the NLI verifier adds a final step that sits below per-span granularity on dev (Section 9). The Hagen et al. 0.440 row is BIO-tag macro F_1 on Hagen's Val split rather than span exact-match on dev, so the Δ column is indicative only; likewise our held-out test F_1 (Table 9) cannot be compared cleanly to Hagen's number because Hagen releases no test split, so the apparent gap mixes splits and metrics.

Table 6
Extraction dev progression. All numbers in this table except the first row are span-level exact-match F_1 on the CCNC dev split (overlap F_1 in parentheses). †The Hagen RoBERTa baseline is BIO-tag macro F_1 on Hagen's evaluation split (Hagen et al. Table 7; their Table 6a labels this as Val, with no separate test column), not span exact-match on dev; the two metrics and the two splits differ, so Δ is an indicative reference, not a directly comparable gain.

Stage	Exact F_1 (overlap F_1)	Δ exact
Hagen RoBERTa baseline [1] (Val, BIO-tag)	0.440 (n/a)	ref†
Single seed, base recipe (seed 123)	0.5742 (0.7692)	+13.4 pp
Hard span-level vote, matched three-seed panel	0.5781 (0.7634)	+13.8 pp
Emission soft-vote, same matched panel	0.5888 (0.7730)	+14.9 pp
Emission soft-vote, production panel with TAPT seed	**0.5933** (0.7800)	**+15.3** pp
+ NLI span verifier ($\tau = 0.1$, within-noise)	0.5953 (0.7781)	+15.5 pp

7. Polarity Classification

7.1. Architecture

We use `bert-base-uncased` [21] with the default sequence-classification head (Figure 5). The one architectural change is to rewrite the bare entity tags as PURE typed entity markers [24]: `<e0>X</e0>` becomes `[E0:CAUSE] X [/E0:CAUSE]` and `<e1>Y</e1>` becomes `[E1:EFFECT] Y [/E1:EFFECT]`. Four special tokens are registered, the embedding table is resized, and the marker embeddings are learned from scratch during fine-tuning. Typed markers contribute a 0.8 pp macro F_1 gain on dev under the same recipe.

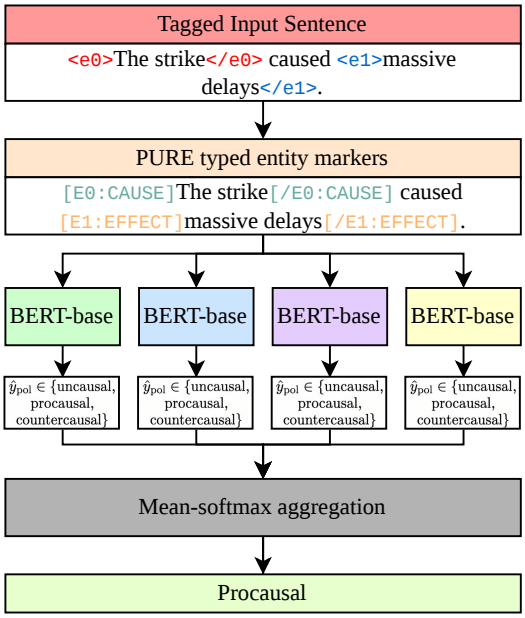

Figure 5: Polarity stack. The tagged input sentence is rewritten with PURE typed entity markers and classified by four independently seeded BERT-base models; their softmax outputs are averaged (mean-softmax) to produce the final three-way label.

7.2. The nine-pattern counter-causal augmentation

The main contribution for polarity is on the training-data side: we generate counter-causal training rows from a large language model conditioned on nine patterns of counter-causal expression adapted from Hagen et al. [1] Table 4: *direct_negation, negated_context, lack_of_counterfactuality, lack_of_effect, implicit_lack_of_effect, negative_causation, usual_inverse_effect, coincidence, temporal_order.* Seven patterns instantiate Hagen Table 4 categories one-to-one ((h) direct negation, (i) negated context, (b) violation of counterfactuality, (c) lack of effect, (f) negative causation, (e) usual inverse effect, (a) violation of temporal order). The remaining two refine surface realizations Hagen lists as examples within existing categories rather than splitting them into separate categories: *coincidence* corresponds to Hagen's (b) example expression "A and B happened coincidentally" and *implicit_lack_of_effect* corresponds to Hagen's (c) implicit example "He is happy. He did not win the lottery." Hagen categories (d) inverse expected cause and (g) contradiction were not implemented; (d) is present in Hagen Table 6c's empirical distribution but its surface form overlaps (c) and (e), while (g) is absent from Table 6c. Hagen et al. explicitly assign *negative_causation* (A *prevents* B) to the *uncausal* class rather than the counter-causal class (their Section 3.1, paragraph below Table 4: "ontological asymmetry (f) … indicates uncausality rather than countercausality"), and (f) does not appear in the empirical distribution of counter-causal expressions in CCNC (their Table 6c). We nevertheless retain (f) as a generation pattern because the resulting sentences share the negated-effect surface form that the polarity model must disambiguate from true counter-causal cases; this is a deliberate decision to inject hard surface-form confusables into the counter-causal class and is a known label-noise risk discussed in Section 10.

For each pattern, a one-shot prompt with a single fixed exemplar and an in-domain seed sentence from CCNC train produces candidates, which are then mechanically validated for byte-equal preservation of the `<e0>...</e0>` and `<e1>...</e1>` content and for the presence of exactly one `<e0>` pair and one `<e1>` pair. The structural-validator acceptance rate is 93.2% (2,433 of 2,610 attempts on the production doubling pass; the smaller pilot pass was similar). Because the validators check only tag preservation and pair-count, this rate is an upper bound on usable yield rather than a semantic quality measure. A doubling augmentation pass grows the counter-causal training class from 886 to 3,319 rows. A seed-42 single-run dev ablation isolates the augmentation effect; each run is checkpointed at the epoch that scores best on a chosen validation metric. Augmentation raises dev macro F_1 from 0.766 (no

augmentation, checkpoint selected by lowest validation loss) to 0.890 (two passes, checkpoint selected by highest validation macro F_1, the criterion the deployed models use). Because these endpoints were checkpointed under different criteria, the controlled comparison is one pass against two under matched F_1 selection, where the second pass adds $+2.4$ pp. Each cell is a single fine-tuning run at seed 42, and the gain is of the same order as the per-seed noise band reported with the ensemble results below, so we treat it as direction-consistent rather than statistically certified. The deployed four-seed ensemble is built from the two-pass-augmented models (Table 7).

7.3. Four-seed ensemble (with offline calibration)

We train four BERT-base models with seeds $\{42, 123, 2026, 7\}$ under a shared recipe, early-stopping each on dev macro F_1 (hyperparameters in Appendix A). For calibration analysis, each model's softmax is rescaled by a single per-model temperature T fitted by LBFGS on the development set to minimize negative log-likelihood, following Guo et al. [25]. Fitted temperatures fall in $T \in [2.18, 2.39]$, and per-model calibration substantially reduces both the negative log-likelihood and the expected calibration error. Because the four models share architecture and training recipe, temperature scaling shifts the ensemble argmax only on a handful of borderline cases and does not affect the ensemble dev macro F_1 (Table 7). The deployed system (Section 8) therefore runs with $T = 1.0$; the per-model temperatures are an offline analysis on the development split.

7.4. Results

Table 7
Polarity dev results. Per-class F_1 on the right. For the production ensemble row, only the macro and counter-causal F_1 are reported: the stored uncausal and procausal per-class numbers are rounded to three decimals and do not reproduce the full-precision macro F_1 of 0.9029 when averaged at that precision.

Configuration	Macro F_1	uncausal F_1	procausal F_1	counter-causal F_1
Seed 42	0.8897	0.9275	0.8678	0.8739
Seed 123	0.8773	0.9091	0.8319	0.8908
Seed 2026	0.8805	0.9246	0.8468	0.8703
Seed 7	0.8867	0.9216	0.8571	0.8814
4-seed mean-softmax (production)	**0.9029**	—	—	**0.891**
5-seed mean-softmax (adds a weaker seed)	0.8953	—	—	—

Table 7 reports the dev results. The four-seed ensemble macro F_1 of 0.9029 sits one to two points above the best single seed and the four-seed mean (per-seed scores in Table 7; four-seed mean 0.8836, sample standard deviation 0.0057, an estimate that is itself uncertain at $n = 4$). Without a held-out distribution of four-seed ensembles, we report the gain as a point improvement rather than a statistically certified one. Adding a fifth seed with a weaker single-seed macro F_1 of 0.873 reduced the ensemble score (Table 7), so we cap the ensemble at four members. This cap, the dev-fit temperatures, and the dev-set seed pool are all decisions made on the same evaluation split with no held-out validation. For context, Hagen et al. [1] Table 7 reports a RoBERTa baseline (size unspecified) for the entity-pair Identification task at macro F_1 0.921 on their Val split (no separate test split is released; see their Table 6a). Their task evaluates candidate entity pairs with labels {no-relationship, causal, countercausal}, whereas our polarity task evaluates dev pairs with labels {procausal, countercausal, uncausal}, so the two numbers are reference points rather than a head-to-head comparison. Our four-seed BERT-base ensemble reaches 0.9029 on dev and 0.817 on test. Two single-seed RoBERTa-large polarity pilots reached macro F_1 0.717 without augmentation and 0.726 with a 1,246-row pilot-era nine-pattern augmentation; the augmented pilot failed in a wrong-direction-bias mode (high counter recall with low counter precision) rather than classical overfitting, and we did not pursue RoBERTa-large further. Closing this gap is the most important open question for our polarity system.

Confusion matrix. The seed-42 confusion matrix on 502 dev sentences (Table 8) shows that the largest residual error class is the 21 uncausal sentences predicted as either procausal or countercausal. Mutual confusion between procausal and countercausal is small, at five to six mistakes per direction; the no-augmentation ablation in Section 7.2 shows that this gap widens without the nine-pattern data.

Table 8
Polarity seed-42 confusion matrix on the dev split. Rows are gold; columns are predicted. Recall on the right.

Gold ↓ / Predicted →	uncausal	procausal	counter-causal	recall
uncausal (264)	243	12	9	0.920
procausal (120)	8	106	6	0.883
counter-causal (118)	8	5	105	0.890

8. Held-Out Test-Set Results

The Touché organizers evaluated each submission on a held-out CCNC test split using the official TIRA [26] evaluator. Table 9 reports the test-set scores for our three subtask submissions alongside the matched development numbers from the per-subtask sections.

Table 9
Held-out CCNC test-set results, with matched dev numbers. Rows marked TIRA are from the official TIRA [26] evaluator; the "final, causal-only" extraction rows are from the organizers' post-deadline re-evaluation of the extraction subtask on the causal test samples only, which raises scores without reordering submissions and will be reported in the lab overview [4]. For extraction the TIRA evaluator reports exact-match F_1 together with a token-level IoU; the dev overlap F_1 is a different relaxation of exact match and is not directly comparable. Δ is test − dev where metrics align. Row-internal arithmetic does not reconcile: the TIRA extraction F_1 (0.473) is a per-sentence F_1 averaged over the test set while the displayed precision (0.475) and recall (0.490) are corpus-level micro values, whose harmonic mean would be 0.482; the polarity macro F_1 (0.817) is the mean of per-class F_1 scores rather than the harmonic mean of macro precision (0.796) and macro recall (0.851), which would be 0.823. We report values as published.

Subtask	Metric	Dev	Test	Δ
Detection (binary)	F_1 (causal class)	0.8946	0.869	−2.6 pp
	precision	—	0.848	—
	recall	—	0.890	—
	accuracy	—	0.832	—
Extraction (TIRA)	exact-match F_1	0.5933	0.473	−12.0 pp
	precision	—	0.475	—
	recall	—	0.490	—
	token IoU	—	0.451	—
	granularity	—	0.583	—
	granularity-adjusted F_1	—	0.453	—
Extraction (final, causal-only)	granularity-adjusted F_1	—	0.728	—
	precision	—	0.764	—
	recall	—	0.787	—
Polarity (3-class)	macro F_1	0.9029	0.817	−8.6 pp
	macro precision	—	0.796	—
	macro recall	—	0.851	—

The dev-to-test gap orders as extraction (12.0 pp exact F_1) > polarity (8.6 pp macro F_1) > detection (2.6 pp), which is consistent with the dev-set overfitting disclosed in Section 10: the soft-vote panel composition and the four-vs-five-seed polarity cap were both selected on dev, whereas detection's only dev-selected hyperparameter is the cross-task stacker rule (the BERT detector itself trains for a

fixed epoch budget with no early stopping). Within extraction, precision and recall are close, while the token-level IoU indicates that on average predicted and gold token sets overlap less than half of their union (Table 9); this aggregate mixes boundary misalignment with full span misses and cannot by itself separate the two. We do not compare the test token IoU to the dev overlap F_1 because the two are different relaxations of exact match. The gaps may also reflect domain shift on the unseen test split, but we have no evidence for a specific shift and do not claim one.

In the organizers' final evaluation across the two participating teams and the organizers' baseline [4], our system scores highest on extraction (granularity-adjusted F_1 0.728, against 0.665 for the baseline and 0.587 for the other team) and highest among the teams on polarity (macro F_1 0.817 against 0.778, with the baseline at 0.839). On detection it ranks third on F_1 (0.869 against 0.884 and 0.872), with the highest recall (0.890) and the lowest precision of the three systems.

9. Discussion and Future Work

Statistical strength of the NLI gain. The post-hoc NLI verifier moves dev exact F_1 from 0.5933 to 0.5953 ($+0.0020$, at a -0.0019 overlap-F_1 cost; Table 6), within noise, and we did not run a paired bootstrap or McNemar test; we therefore treat the gain as direction-consistent rather than statistically certified. The qualitative evidence agrees: all three spans dropped at $\tau = 0.1$ are truncated fragments of gold spans, false positives under the exact metric that still carried partial overlap credit, which is why removing them helps exact F_1 while slightly lowering overlap F_1 (Section 6.3).

Future work. Two threads follow from the present results. The first targets polarity supervision by adding a rule-based negation-scope feature to the polarity model, augmenting the input with one extra per-token label ("in negation scope?"); this is motivated by the 67.8% counter-causal versus 15.8% procausal negation-cue prevalence gap documented in Section 3. The second targets the extraction backbone and candidate generation. Alternative encoders include ELECTRA-large, structurally distinct from RoBERTa's MLM objective; XLM-RoBERTa-large [27], pretrained on CC-100 across 100 languages and potentially better suited to Indian-English news register; and NER architectures that decouple span-boundary modeling from interior tagging, whether through a locate-then-label boundary-regression stage [28], word-pair relation classification [29], or biaffine span scoring [30]. A separate direction restricts extraction-stage span candidates to constituent edges from a parser, which would render many one-sided clause-attachment errors structurally impossible when the parser is correct.

10. Limitations

We disclose the principal threats to validity rather than relegate them to a footnote. The body of Sections 5–7 reports CCNC development numbers; the held-out TIRA test scores are lower on every subtask (Table 9), and the comparison against Hagen et al. assumes their exact-match definition matches ours, which we have not independently verified. Nearly every discrete design choice, from the soft-vote panel and the stacker rule to the NLI threshold, the augmentation dose, the input-marker format, and the data-inclusion decisions, was selected on the same development split with no held-out internal validation; the 12.0 pp dev-to-test gap on extraction, the largest of the three, is the most plausible aggregate signature of this selection.

Statistical support for the smaller gains is correspondingly thin. The post-hoc NLI verifier yields a within-noise exact-F_1 gain on dev, which we report as direction-consistent rather than certified (Section 9). Per-seed variance is available only for the polarity ensemble (mean 0.8836, sample standard deviation 0.0057 over four seeds); the soft-vote gain on extraction and the stacker gain on detection are single-run point estimates without variance bands.

CCNC, drawn from Indian-English news outlets, is the only corpus we evaluate on, so transfer of the soft-vote and augmentation findings to other domains, English varieties, or non-news genres is untested. Finally, although the augmentation prompts draw exclusively on the CCNC training split, so

held-out sentences cannot enter the training data by construction, GPT-5.4 may have seen CCNC or Hagen et al.'s pattern exemplars during pretraining; we did not run overlap checks of the generated outputs against the dev or test splits, and the structural validators certify tag preservation, not semantic counter-causality.

11. Conclusion

We described a system for the Touché 2026 Causality Extraction task on the Countercausal News Corpus. On the held-out test set it reaches binary F_1 0.869 on detection and macro F_1 0.817 on polarity, and in the organizers' final causal-only evaluation of extraction it scores granularity-adjusted F_1 0.728, the highest extraction score among all submissions including the organizers' baseline (Section 8). The development split, used for component selection and ablations, scores higher throughout (Table 9), including exact-match extraction F_1 0.5933 against 0.473 under TIRA's all-samples exact-match metric on test, which a post-hoc DeBERTa-MNLI span verifier raises further within the noise band on dev (Section 9).

The system combines a three-layer BILOU+CRF stack on RoBERTa-large with emission-level soft-vote ensembling across three seeds, a single-rule cross-task stacker in which extraction-stage spans gate detection-stage false positives, and a nine-pattern counter-causal augmentation that adds 2,433 polarity training examples. The contribution is empirical rather than architectural: to our knowledge, neither emission-level CRF soft-voting with matched-panel measurement on the Causal News Corpus family nor a Hagen-style counter-causal pattern taxonomy applied to LLM-based augmentation on CCNC has been reported before. The single-rule cross-task stacker is the cross-task interaction we have the most evidence on: it added 2.6 pp on dev detection causal-class F_1, but the dev-to-test drop on detection is of the same magnitude, so we cannot attribute net transfer to the rule without an unstacked test baseline. We retain it because precision-recall asymmetry favors precision and because the stacker did not degrade performance on any split we tested.

Acknowledgments

We thank the Touché organizers for the lab and the evaluation infrastructure [26], and Hagen et al. [1] for releasing both the CCNC corpus and a public baseline. Compute was provided by personal hardware and by University of Zurich shared GPU resources.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI GPT-5.4 for data generation, producing the nine-pattern counter-causal training examples for the polarity task described in Section 7.2 under the structural validation reported there, and Claude (Anthropic) for code-generation assistance on training and evaluation scripts, which were human-reviewed and tested, and for writing assistance (grammar and spelling check, paraphrase and reword). All scientific decisions, analyses, and conclusions are the authors' own. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] T. Hagen, N. Deckers, F. Wolter, H. Scells, M. Potthast, Investigating counterclaims in causality extraction from text, arXiv preprint arXiv:2510.08224 (2025). URL: https://arxiv.org/abs/2510.08224.

[2] 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, arXiv preprint arXiv:1907.11692 (2019). URL: https://arxiv.org/abs/1907.11692.

[3] 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. de 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.

[4] 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. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[5] F. A. Tan, A. Hürriyetoğlu, T. Caselli, N. Oostdijk, T. Nomoto, H. Hettiarachchi, I. Ameer, O. Uca, F. F. Liza, T. Hu, The causal news corpus: Annotating causal relations in event sentences from news, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 2298–2310. URL: https://aclanthology.org/2022.lrec-1.246/.

[6] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, T. Caselli, O. Uca, F. F. Liza, N. Oostdijk, Event causality identification with causal news corpus — shared task 3, CASE 2022, in: Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2022), 2022. URL: https://arxiv.org/abs/2211.12154.

[7] T. P. Schrader, S. Razniewski, L. Lange, A. Friedrich, BoschAI @ causal news corpus 2023: Robust cause-effect span extraction using multi-layer sequence tagging and data augmentation, in: Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2023), 2023. URL: https://aclanthology.org/2023.case-1.5/.

[8] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, N. Oostdijk, T. Caselli, T. Nomoto, O. Uca, F. F. Liza, S.-K. Ng, RECESS: Resource for extracting cause, effect, and signal spans, in: Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023), Volume 1: Long Papers, Nusa Dua, Bali, 2023, pp. 66–82. URL: https://aclanthology.org/2023.ijcnlp-main.6/. doi:10.18653/v1/2023.ijcnlp-main.6, also referred to in the literature as Causal News Corpus v2 (CNCv2).

[9] J. Wei, K. Zou, EDA: Easy data augmentation techniques for boosting performance on text classification tasks, in: Proceedings of EMNLP-IJCNLP 2019, 2019. URL: https://aclanthology.org/D19-1670/.

[10] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, N. Oostdijk, O. Uca, S. Thapa, F. F. Liza, Event causality identification — shared task 3, CASE 2023, in: Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2023), INCOMA Ltd., Varna, Bulgaria, 2023, pp. 144–150. URL: https://aclanthology.org/2023.case-1.19/.

[11] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the 18th International Conference on Machine Learning (ICML 2001), 2001.

[12] L. Ratinov, D. Roth, Design challenges and misconceptions in named entity recognition, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009), 2009. URL: https://aclanthology.org/W09-1119/.

[13] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of NAACL-HLT 2016, 2016. URL: https://aclanthology.org/N16-1030/.

[14] X. Dai, H. Adel, An analysis of simple data augmentation for named entity recognition, in: Proceedings of COLING 2020, 2020. URL: https://aclanthology.org/2020.coling-main.343/.

[15] T. Schick, H. Schütze, Generating datasets with pretrained language models, in: Proceedings of EMNLP 2021, 2021. URL: https://aclanthology.org/2021.emnlp-main.555/.

[16] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, in: Proceedings of EMNLP-IJCNLP 2019, 2019. URL: https://aclanthology.

org/D19-1404/.

[17] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: Proceedings of the International Conference on Learning Representations (ICLR 2023), 2023. URL: https://arxiv.org/abs/2111.09543.

[18] M. Laurer, W. van Atteveldt, A. Casas, K. Welbers, DeBERTa-v3-large-mnli-fever-anli-ling-wanli: a zeroshot classifier, 2022. URL: https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli, huggingFace model card.

[19] A. Williams, N. Nangia, S. R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (NAACL-HLT 2018), 2018. URL: https://aclanthology.org/N18-1101/.

[20] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don't stop pretraining: Adapt language models to domains and tasks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), 2020. URL: https://aclanthology.org/2020.acl-main.740/.

[21] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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 (NAACL 2019), 2019. URL: https://aclanthology.org/N19-1423/.

[22] F. A. Tan, X. Zuo, S.-K. Ng, UniCausal: Unified benchmark and repository for causal text mining, in: Big Data Analytics and Knowledge Discovery (DaWaK 2023), Lecture Notes in Computer Science, Springer, 2023. URL: https://arxiv.org/abs/2208.09163.

[23] E. Zhu, J. Li, Boundary smoothing for named entity recognition, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022), 2022. URL: https://arxiv.org/abs/2204.12031.

[24] Z. Zhong, D. Chen, A frustratingly easy approach for entity and relation extraction, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2021), 2021. URL: https://arxiv.org/abs/2010.12812.

[25] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: Proceedings of the 34th International Conference on Machine Learning (ICML 2017), 2017. URL: https://arxiv.org/abs/1706.04599.

[26] 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.

[27] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747/.

[28] Y. Shen, X. Ma, Z. Tan, S. Zhang, W. Wang, W. Lu, Locate and label: A two-stage identifier for nested named entity recognition, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), 2021. URL: https://arxiv.org/abs/2105.06804.

[29] J. Li, H. Fei, J. Liu, S. Wu, M. Zhang, C. Teng, D. Ji, F. Li, Unified named entity recognition as word-word relation classification, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2022), 2022. URL: https://arxiv.org/abs/2112.10070.

[30] 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 (ACL 2020), 2020. URL: https://arxiv.org/abs/2005.07150.

A. Training Hyperparameters

Table 10 lists the fine-tuning configuration of the three deployed models. The TAPT encoder used by extraction seed 42 is produced beforehand by 50 epochs of masked-language modeling on the 4,327-sentence in-domain corpus (learning rate 10^{-4}, 15% token masking). Full configuration files ship with the released repository.

Table 10
Fine-tuning hyperparameters per subtask model. Cells marked — do not apply; unlisted settings follow library defaults.

Parameter	Detection	Extraction (per seed)	Polarity (per seed)
Encoder	BERT-base	RoBERTa-large	BERT-base
Optimizer	AdamW	AdamW ($\varepsilon = 10^{-6}$)	AdamW
Learning rate	2×10^{-5}	5×10^{-5}	2×10^{-5}
CRF-group learning rate	—	3×10^{-4}	—
Weight decay	0.01	0.01	—
Schedule	10% linear warmup, cosine	6% warmup, cosine	—
Gradient clip	1.0	1.0	—
Batch size	32	16	32
Max sequence length	128	256	256
Dropout	model default	0.1	model default
Epochs	5 (fixed)	$\leq$30	$\leq$12
Early stopping	none	patience 5, overlap F_1	patience 4, macro F_1
Seeds	42	123, 2026, 42	42, 123, 2026, 7
EDA (α_{sr} / α_{ri})	—	0.4 / 0.5	—
Aux. label smoothing	—	$\varepsilon = 0.1$, weight 0.2 (seed 42)	—

B. Reproducibility Details

Compute and release. All models were fine-tuned on a single NVIDIA RTX 5090 (32 GB); the seven deployed seeds complete in under twelve GPU-hours, and full three-subtask inference on the development split runs in under sixty seconds. The stack is PyTorch 2.4.1 (CUDA 12.1), transformers 4.46.3, tokenizers 0.20.3, and pytorch-crf 0.7.2, with full version pins in the released repository at https://github.com/rzninvo/NegaCausal, which also hosts the augmentation prompts and generated counter-causal data, the inference and soft-vote code, the TIRA Docker recipe, the per-seed checkpoints, and the stacker calibration; it will be tagged at the CEUR camera-ready version. CCNC is distributed by the Touché organizers under the lab data-use agreement and used only for CLEF 2026 participation; CNCv2/RECESS [8] and UniCausal [22] are used train-split-only under their upstream licenses, the DeBERTa-v3 NLI checkpoint [18] under its MIT license, and we redistribute none of these.

Counter-causal data generation. The 2,433 augmented examples were generated by OpenAI's GPT-5.4 via the Chat Completions API (JSON-response mode, temperature 0.9, top_p 1.0, max_completion_tokens 300); the snapshot identifier and per-pattern prompt templates are in the repository. Seeds and the single fixed exemplar come exclusively from the CCNC training split, so no dev or test sentence appears in any prompt. Each one-shot prompt rewrites an in-domain seed into the target pattern while preserving the `<e0>` and `<e1>` span contents verbatim; candidates are validated for byte-equal tag preservation and exactly one cause/effect pair (93.2% acceptance, 2,433 of 2,610 attempts, a syntactic rather than semantic pass rate).