Overview of Touché 2026 — Argumentation Systems

Extended Version

Johannes Kiesel[1], Marc Feger[2], Tim Hagen[3,7], Sebastian Heineking[3], Maximilian Heinrich[4], Maik Fröbe[5], Katarina Boland[2], Christine Mathews[4], Wilhelm Pertsch[5], Julia Romberg[1], Ines Zelch[5], Stefan Dietze[1,2], Matthias Hagen[5], Martin Potthast[3,6,7] and Benno Stein[4]

[1]*GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany*
[2]*Heinrich Heine University Düsseldorf, Düsseldorf, Germany*
[3]*University of Kassel, Kassel, Germany*
[4]*Bauhaus-Universität Weimar, Weimar, Germany*
[5]*Friedrich-Schiller-Universität Jena, Jena, Germany*
[6]*hessian.AI, Darmstadt, Germany*
[7]*ScaDS.AI, Dresden, Germany*

Abstract

This paper is the extended overview of Touché: the seventh edition of the lab on argumentation systems that is held at CLEF 2026. Motivated by threats to the information ecosystem and thus to democratic societies, the Touché lab continues to investigate supportive technologies for decision-making and opinion-forming. This year, we organized four shared tasks, of which three are new: (1) Fallacy Detection, in which participants detect and classify fallacies and argument schemes in short texts (new task); (2) Causality Extraction, in which participants detect and classify causal information in short texts (new task); (3) Generalizability of Argument Identification in Context, in which participants classify whether a sentence, given provenance data, was annotated as being an argument or not (new task); and (4) Advertisement in Retrieval-Augmented Generation, in which participants detect and remove ads from search responses (2nd edition). In this paper, we describe these tasks, their setup, and participating approaches in detail.

Keywords

Advertisement Detection, Argument Identification, Computational Argumentation, Fallacy Detection, Generalizability.

1. Introduction

Decision-making and opinion-forming are everyday tasks. With the ubiquity of web search and language models, it is easy to find arguments on any topic. However, while the systems of today are quick to produce an answer, they rarely provide transparent quality assurance of arguments and might even misuse their position as intermediary to manipulate information for commercial gains. In this context, the Touché lab series, running since 2020, has several tasks to advance both argumentation systems and the evaluation thereof. The data—including judgments—and publications of past and current tasks

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
johannes.kiesel@uni-weimar.de (J. Kiesel); marc.feger@hhu.de (M. Feger); tim.hagen@uni-kassel.de (T. Hagen); sebastian.heineking@uni-kassel.de (S. Heineking); maximilian.heinrich@uni-weimar.de (M. Heinrich); maik.froebe@uni-jena.de (M. Fröbe); katarina.boland@hhu.de (K. Boland); christine.mathews@uni-weimar.de (C. Mathews); wilhelm.pertsch@uni-jena.de (W. Pertsch); julia.romberg@gesis.org (J. Romberg); ines.zelch@uni-jena.de (I. Zelch); stefan.dietze@gesis.org (S. Dietze); matthias.hagen@uni-jena.de (M. Hagen); martin.potthast@uni-kassel.de (M. Potthast); benno.stein@uni-weimar.de (B. Stein)
0000-0002-1617-6508 (J. Kiesel); 0009-0008-6385-1722 (M. Feger); 0009-0000-4854-7249 (T. Hagen); 0000-0002-7701-3294 (S. Heineking); 0000-0001-5450-8203 (M. Heinrich); 0000-0002-1003-981X (M. Fröbe); 0000-0003-2958-9712 (K. Boland); 0009-0006-3041-0002 (C. Mathews); 0009-0008-9232-4788 (W. Pertsch); 0000-0003-0033-9963 (J. Romberg); 0009-0005-2659-5326 (I. Zelch); 0009-0001-4364-9243 (S. Dietze); 0000-0002-9733-2890 (M. Hagen); 0000-0003-2451-0665 (M. Potthast); 0000-0001-9033-2217 (B. Stein)

are available on our web page.[1] This paper is the extended version of [1] and describes the 2026 edition of Touché, which featured the following shared tasks:

1. Fallacy Detection (new task, cf. Section 2) features three sub-tasks for argument type detection, in which participants classify (1) whether an argument is fallacious, (2) the type of a fallacy, and (3) the scheme class of an argument. 12 teams participated and submitted 51 runs.
2. Causality Extraction (new task, cf. Section 3) features three sub-tasks, in which participants (1) classify whether a text span contains causal information, (2) mark the respective entities in the span, and (3) classify the relation between two marked entities and procausal, concausal, or uncausal. 2 teams participated and submitted 6 runs.
3. Generalizability of Argument Identification in Context (new task, cf. Section 4), in which participants classify whether a sentence, in its context and with provenance data, constitutes an argument or not. 4 teams participated and submitted 8 runs.
4. Advertisement in Retrieval-Augmented Generation (2nd edition, cf. Section 5) features three sub-tasks in countering advertisements in the output of LLMs, in which participants (1) classify whether a response contains an advertisement or not, (2) locate the advertisement in a response, and (3) remove the advertisement from the response without affecting the coherence of the response. 3 teams participated and submitted 9 runs.

Submission via TIRA As in the previous editions of Touché, we used TIRA [2] as the submission platform. Participants could either submit code, software, or run files.[2] To also enable efficiency-oriented evaluations of code submissions, we used the TIREX Tracker [3] to monitor the resource consumption (energy usage and GPU/CPU/RAM usage) of approaches executed in TIRA. We encouraged code and software submissions to increase reproducibility, as approaches submitted this way can later be executed again, including on different data of the same format. For code and software submissions, teams implemented their approach as a Docker image and either uploaded it to their dedicated Docker registry in TIRA (software submission) or submitted a git repository from which the TIRA client builds a Docker image (code submission). Code submission ensure that the git repository is clean (i.e., all changes are committed and no untracked files), which allows linking a Docker image to the exact version of a git repository that produced a submission, whereas software submissions do not need to be linked to the git repository. Submissions in TIRA are immutable, and a team could upload as many code or software submissions as they liked; only they and TIRA had access to their dedicated Docker image registry (i.e., the images were not public while the shared task was ongoing). To improve reproducibility, TIRA executes software in a sandbox by removing the internet connection (ensuring that the software is fully installed in the Docker image, which eases rerunning software later, as libraries and models must be installed in an image). For the execution, participants could select the resources that their software had available for execution, from 4 CPU cores with 40 GB RAM up to 5 CPU cores with 50 GB RAM and 1 Nvidia A100 GPU with 40 GB RAM. Participants could run their software multiple times using different resources to study the scalability and reproducibility (e.g., whether the software executed on a GPU yields the same results as on a CPU). TIRA used a Kubernetes cluster with 1,620 CPU cores, 25.4 TB RAM, and 4 A100 GPUs to schedule and execute the software submissions and allocate the resources that the participants selected.

Lab Registrations For the seventh edition of the Touché lab, we received 97 registrations from 25 countries (vs. 62 registrations in 2025). The most lab registrations came from India (16). In total, 20 of the registered teams participated actively. Active teams in previous editions of Touché were: 12 in 2025, 20 in 2024, 7 in 2023, 23 in 2022, 27 in 2021, and 17 in 2020.

The following sections detail each task, the participants' submissions, and the outcomes.

[1] https://touche.webis.de/
[2] https://tira.io

2. Fallacy Detection

Arguments serve to exchange reasons and justify claims, yet not every argument that appears valid is in fact valid; such arguments are called fallacies [4], motivating the tasks of fallacy detection and classification. A complementary perspective on arguments comes from argument schemes, which capture recurring patterns of reasoning [5]. In principle, the two notions are closely linked, as a fallacy can often be understood as a defective or misleading instance of an otherwise legitimate scheme. In practice, however, fallacy detection, fallacy classification, and argument scheme classification each pose substantial challenges on their own. We therefore operationalize all three as separate classification tasks, rather than modeling explicitly how schemes and fallacies interact.

Overview The task consists of three sub-tasks: (1) Fallacy Detection: Given an argument, determine whether it is fallacious. (2) Fallacy Classification: Given a fallacious argument, identify the specific type of fallacy. (3) Argument Scheme Classification: Given a non-fallacious argument, determine which argumentation scheme it follows.

Each sub-task is offered in two evaluation tracks. In the *base* track, submitted systems see only the original argument as a self-contained comment; in the *enhanced* track, they may additionally use a rewritten version in which the implicit reasoning and underlying assumptions are made explicit. The two tracks are evaluated separately to measure the utility of the explicit rewrite.

2.1. Background

What counts as a fallacy We adopt the working definition underlying the dataset of Sahai et al. [6], which we build on: a fallacy is an argument that appears convincing at first glance but is flawed—either because its premises or evidence are incorrect or insufficient, or because the inference from the premises to the claim is too weak or otherwise unjustified.

Argument schemes and their dimensions An argument consists of a claim together with one or more premises intended to support it. Arguments typically instantiate recurring inference patterns known as argument schemes—for example, the argument from expert opinion, in which an expert's assertion supports a claim [5]. Rather than working with the full inventory of individual schemes, we build on the more abstract dimensions of Macagno [7, 8]. Individual schemes can be mapped onto these dimensions, which therefore offer a coarser but more general characterization of an argument's reasoning type. This characterization applies to any argument, so both fallacious and non-fallacious arguments can be described in the same terms, though for symmetry with fallacy classification we restrict scheme classification (sub-task 3) to non-fallacious arguments. Strictly speaking, the resulting classes are not individual argument schemes, but combinations of these broader dimensions.

Computational fallacy detection Most work treats fallacy detection as a classification task and mainly improves *how* the argument is represented: by enriching the model input with retrieved examples or generated context [9, 10], by encoding argumentative structure in the model architecture [11, 12], or by translating arguments into first-order logic for a neuro-symbolic solver [13]. A closely related task is propaganda-technique detection, whose techniques often overlap with argumentative fallacies [14, 15]. A second line of work moves beyond label prediction to the inferential relation between premises and claims: Ruiz-Dolz et al. show that fallaciousness depends on whether an argument scheme's critical questions are answered, not on surface patterns [16], while Glockner et al. reconstruct the implicit premises linking misleading claims to cited evidence, together with the fallacy underlying that inferential link [17].

2.2. Data Description

The dataset is drawn from a curated subset of the Reddit-based informal fallacies dataset of Sahai et al. [6]. It comprises 1,171 arguments (938 train, 233 test) covering eight fallacy types, approximately balanced between 580 fallacious and 591 non-fallacious arguments. Each non-fallacious argument was deliberately selected to structurally resemble a specific fallacy type, making detection substantially harder than separating fallacies from arbitrary non-fallacious text. For argument scheme classification, we operationalize scheme class as one of four goal–basis combinations derived from Macagno's means–end dimensions of argumentation schemes [7, 8]: the argument's *goal* (practical or epistemic) and its *basis* (internal or external).

We provide each argument in several variants. Since Reddit comments are often enthymematic and rely on surrounding context, the *base* text reconstructs the raw comment into a more self-contained, understandable argument by integrating the parent comment and thread title. The *enhanced* text is a second rewrite of the base argument, this time using knowledge of its underlying argumentation scheme and possible fallacy: for every argument, the two scheme dimensions are made explicit, and for fallacious arguments, the fallacy type is surfaced as well. For both views, we also extract the argument structure as a claim and its premises.

2.3. Participant Approaches

In 2026, 12 teams participated, submitting 51 runs in total: 26 to sub-task 1, 18 to sub-task 2, and 7 to sub-task 3, complemented by an organizer baseline run on each track per sub-task. Most teams relied either on computationally efficient encoder fine-tuning or on a focused use of large language models. Several teams systematically compared the base and enhanced inputs and found that the enhanced variant substantially improves results. Others further investigated what these gains actually measure: Team helium, Team UZHCL_for_the_win, and Team Veritas argued that the enhanced rewrite carries label-bearing signal by design, so that part of the higher score reflects input cues rather than genuine fallacy recognition.

Baseline As an organizer-provided baseline, we apply zero-shot prompting with Llama-3.2-3B to all three sub-tasks on both the base and enhanced inputs. Invalid model outputs are replaced with valid labels chosen uniformly at random.

Team ArguMind [18] For all three sub-tasks, the team's submitted system is a single multi-task DeBERTa-v3-xsmall model with a shared encoder and task-specific heads, which outperforms their TF-IDF LinearSVC pipeline (retained as a strong sparse-feature baseline) on all three sub-tasks.

Team Graz University of Technology (TU Graz) [19] The team contributes a comparative study of prompting strategies rather than a fine-tuning system, evaluating per-class binary prompting, binary prompting with few-shot calibration, and single-prompt multiclass prompting across two LLMs (GPT-5.4-mini and Gemma-4-31B-it) and both data conditions.

Team helium [20] For fallacy detection, the team focuses on evaluation methodology, systematically comparing lexical baselines, dense-embedding classifiers, fine-tuned encoders, and an instruction-tuned language model across raw, base, and enhanced inputs.

Team NapierNLP [21] For fallacy detection, the team uses comparatively small language models and model fusion, fine-tuning four pre-trained small language models—GPT-2, Qwen2.5, TinyLlama, and GPT-Neo—on the task's training data and combining them through confidence-based and majority voting; the best submitted ensembles additionally include fine-tuned BERT and RoBERTa.

Team NIT-Agartala-NLP [22] For fallacy detection and classification, the team uses an ensemble framework of traditional classifiers over TF-IDF features: a stacking ensemble of Logistic Regression, Decision Tree, and Random Forest models for binary detection, and bagging of Logistic Regression for multi-class classification, each with 10-fold cross-validation.

Team OoOoO [23] For fallacy detection, the team averages RoBERTa-base ensembles across seeds and input formats, using 5-fold cross-validation and a single F_1-optimized threshold tuned on the merged out-of-fold predictions. To curb false positives on fallacy-resembling arguments, they move the extracted claim and premises to the front of the input, so the model focuses on the claim–premise relation and the structure survives truncation.

Team Seals-NLP [24] The team compares fine-tuned encoders (BERT, RoBERTa, and ModernBERT) against Qwen3-family LLMs evaluated at several adaptation depths (frozen embedders, SupCon-LoRA adapters, and zero-shot prompting). Their joint encoder architecture predicts fallacy presence, type, and scheme together, trained with token mean-pooling and a masked cross-entropy loss on both the base and enhanced text.

Team Sophisma [25] For fallacy detection and classification, the team fine-tunes a RoBERTa-base model on inputs that combine the enhanced text with explicit claim and premise fields, and contrasts it with a feature-based XGBoost baseline.

Team UZHCL_for_the_win [26] For all three sub-tasks, the team's submitted system is a fine-tuned RoBERTa-large encoder, augmented with synthetic data from LLM-generated contrastive pairs, with additional pseudo-labeling on the enhanced classification and scheme sub-tasks. Alongside it, they run a comparative study that pits several encoders (RoBERTa, DeBERTa, ModernBERT) against Qwen-based LLMs.

Team uzh_tt [27] Across the sub-tasks, the team uses a confidence-adaptive, two-stage system for detection and classification. A fine-tuned RoBERTa model handles high-confidence predictions, while low-confidence cases in fallacy detection and classification are forwarded to GPT-5.4 for further reasoning; the routing thresholds are learned from out-of-fold cross-validation predictions to maximize macro F_1. Scheme classification is handled by RoBERTa alone.

Team Veritas [28] For fallacy detection and classification, the team fine-tunes two encoder-only models, RoBERTa-large and ModernBERT-large (ModernBERT-large only on the classification sub-task), and compares base versus enhanced inputs, finding that a simple bag-of-words classifier already matches the fine-tuned encoders on the enhanced text.

Team Webis [29] The team uses a courtroom-styled, multi-agent pipeline for binary fallacy detection, in which adversarial LLMs act as prosecutor and defense before a neutral judge, while a fine-tuned BERT classifier serves as expert witness and appellate mechanism. It combines retrieval-augmented prompting over role-specific precedent pools with a one-directional appeal mechanism.

2.4. Evaluation Setup

We evaluate all sub-tasks on a held-out test set of 233 arguments. Sub-tasks 2 and 3 are evaluated on their relevant subsets, namely fallacious arguments for fallacy classification and non-fallacious arguments for scheme classification. Our primary metric is macro-averaged F_1, which weights all classes equally and thus keeps rare classes visible. Base and enhanced submissions are scored separately. The results for the three sub-tasks are shown in Tables 1, 2, and 3.

Table 1
Fallacy detection sub-task 1: binary fallacy detection over all 233 test arguments. Each row is one submitted run (*Team*, *Approach*); the *base* and *enhanced* tracks are ranked independently by macro-F_1 and aligned by row position only. Cells show macro-F_1 (lighter is higher); gray rows fall outside the ranking (organizer baseline and informational runs).

base			**enhanced**		
Team	Approach	F_1	Team	Approach	F_1
TU Graz	llm-court-01	0.807	Seals-NLP	Seals-NLP	0.957
UZHCL	st1	0.751	UZHCL	st1-first-run	0.944
Veritas	RoBERTa-large	0.737	OoOoO	enhanced	0.940
helium	Gem4-31B-no-desc	0.736	Veritas	RoBERTa-large	0.927
helium	Gemma4-31TrueRawCtx	0.736	ArguMind	multitask-DeBERTa	0.927
helium	gemma-4-31b-it-context	0.736	Sophisma	RoBERTa-BF16	0.914
NapierNLP	CVB	0.734	NapierNLP	CVB	0.910
OoOoO	base-2	0.713	uzh_tt	cascaded-run-1	0.893
Webis	Webis-CourtroomTrial	0.704	uzh_tt	cascaded-v2	0.893
NIT-Agar.	fine-tuned DeBERTa	0.697	NapierNLP	MVB	0.888
OoOoO	base	0.665	ArguMind	TF-IDF-LinearSVC	0.884
NIT-Agar.	nli-DeBERTa	0.658	uzh_tt	Roberta-Run	0.884
Baseline	baseline	0.460	NIT-Agar.	Stacking TF-IDF	0.875
			Webis	Webis-CourtroomTrial	0.841
			Baseline	baseline	0.438

Table 2
Fallacy detection sub-task 2: 8-way fallacy-type classification over the 115 fallacious arguments. Each row is one submitted run (*Team*, *Approach*); the *base* and *enhanced* tracks are ranked independently by macro-F_1 and aligned by row position only. Cells show macro-F_1 (lighter is higher); gray rows fall outside the ranking (organizer baseline and informational runs). † marks a repaired run (missing or invalid predictions scored as wrong).

base			**enhanced**		
Team	Approach	F_1	Team	Approach	F_1
UZHCL	st2	0.859	uzh_tt	cascaded-V22	0.991
TU Graz	llm-court-02†	0.671	uzh_tt	cascaded-run2-2	0.991
Veritas	RoBERTa-large	0.661	Sophisma	RoBERTa-BF16-Class	0.972
TU Graz	llm-court-01†	0.638	Seals-NLP	Seals-NLP	0.963
Veritas	ModernBERT-lg	0.523	ArguMind	multitask-DeBERTa	0.956
Baseline	baseline	0.050	UZHCL	st2	0.954
			ArguMind	TF-IDF-LinearSVC	0.954
			Veritas	RoBERTa-large	0.947
			uzh_tt	roberta-task2-run	0.931
			Veritas	Ensemble	0.927
			Veritas	ModernBERT	0.927
			NIT-Agar.	Reg-Lr	0.773
			NIT-Agar.	Bagging-LR	0.766
			Baseline	baseline	0.045

On the base track of sub-task 1 (Table 1), the top-ranked submission is Team Graz University of Technology's `llm-court-01` run, a purely prompt-based LLM classifier, with a macro-F_1 of 0.807 [19]. The enhanced track is led by Seals-NLP, whose jointly trained multi-task RoBERTa-large reaches 0.957. For sub-task 2, UZHCL_for_the_win tops the base track with a fine-tuned RoBERTa-large model augmented with synthetic LLM-generated data, while uzh_tt's two-stage system leads the enhanced track. The UZHCL team again ranks first on the base track of sub-task 3, whereas the enhanced track is led by Team ArguMind's multi-task DeBERTa system [26, 18].

Table 3
Fallacy detection sub-task 3: goal/basis scheme classification over the 118 non-fallacious arguments, across the four Macagno classes. Each row is one submitted run (*Team*, *Approach*); the *base* and *enhanced* tracks are ranked independently by macro-F_1 and aligned by row position only. Cells show macro-F_1 (lighter is higher); gray rows fall outside the ranking (organizer baseline).

base			**enhanced**		
Team	Approach	F_1	Team	Approach	F_1
UZHCL	st3	0.716	ArguMind	multitask-DeBERTa	0.976
TU Graz	llm-court-01	0.700	ArguMind	TF-IDF-LinearSVC	0.880
Baseline	baseline	0.498	uzh_tt	Roberta-run-task3	0.858
			Seals-NLP	Seals-NLP	0.816
			UZHCL	st3	0.748
			Baseline	baseline	0.499

3. Causality Extraction

Many questions in domains such as medicine, economics, and public policy require causal reasoning. Questions in web search, between humans or to LLMs are frequently causal [30, 31] and answering them requires structured causal knowledge, which can be extracted from natural language text through causality extraction. However, existing causality extraction research has largely focused on identifying statements that assert causal relations (*causal claims*), while neglecting statements that refute such relations (*countercausal claims*). This neglect is problematic, since countercausal claims are common due to societal discourse in controversial topics or updated knowledge. Ignoring such counterclaims therefore leads to incomplete or misleading causal knowledge bases, with negative consequences for downstream applications like causal question answering systems.

Overview This task addresses the extraction of (counter-)causal claims from natural language text. Participating teams may submit software for any combination of three sub-tasks that build on each other. First, in causality detection, systems are given a natural language text and classify whether it contains causal information or not. Second, in candidate extraction, systems mark the entities, events, or concepts that participate in a causal or countercausal claim. Third, in causality identification, systems are given a natural language text and two marked text spans, e_0 and e_1, and classify whether the text supports that e_0 causes e_1 (causal), challenges that e_0 causes e_1 (countercausal), or does not make a statement about causality from e_0 to e_1 (uncausal).

3.1. Background

Causality extraction is the task of identifying cause–effect claims expressed in natural language. Prior work has approached the problem either end-to-end or as a pipeline consisting of causality detection, labeling of spans that mark causal events, and relationship identification between candidate event pairs [32]. Early systems relied on manually defined lexical and syntactic patterns for explicit causal signals such as *because* or *due to* [33, 34, 35, 36]. Later work introduced supervised neural models, including convolutional architectures enriched with external knowledge [37], sequence-labeling approaches for cause–effect span extraction [38], and transformer-based classifiers for causal relationship identification [39, 40, 41]. Despite this progress, causality remains challenging since causal relationships are often implicit, lexically diverse, and dependent on semantic and contextual interpretation rather than on surface markers alone [42, 43, 44].

Existing causality extraction datasets cover several domains, including news, biomedical text, finance, political text, and social media [45, 46, 47, 48, 49]. However, they typically distinguish only causal from non-causal instances. As a result, statements that explicitly deny or refute a plausible causal relationship are treated like ordinary non-causal statements, even though they carry important causal

information. The Countercausal News Corpus [50] addresses this gap by separating non-causal text into *countercausal* and *uncausal* cases. This makes the shared task different from previous causality extraction settings. Systems must not only detect and extract causal information, but also recognize when a text rejects a causal link between two marked spans. Countercausal instances are especially difficult because they often contain similar cue words to causal claims but differ by negation, or in strictly semantic cues that inverse the meaning.

3.2. Data Description

All three sub-tasks of this task are evaluated on the Countercausal News Corpus (CCNC) [50], an extension of the Causal News Corpus v2 (CNCv2) [32] for countercausality-aware causality extraction. Whereas traditional causality extraction datasets distinguish only between causal and non-causal statements, CCNC introduces an explicit label for *countercausal* claims: statements that refute, deny, or otherwise challenge a causal relationship. Such statements may assert, for example, that a proposed cause had no effect, that two events merely coincided, or that the available evidence does not support a causal link.

CCNC distinguishes three classes: *causal*, *countercausal*, and *uncausal*. Causal statements express that one event or concept causes another; countercausal statements express that a plausible causal relationship is refuted; and uncausal statements do not express a causal or countercausal relationship. The corpus comprises 3415 texts in total: 1028 causal claims, 952 countercausal claims, and 1435 uncausal statements. It was constructed by using CNCv2 as a starting point, manually rewriting a subset of causal sentences into countercausal variants, and reannotating the resulting corpus according to extended annotation guidelines for countercausality [50, Figure 4]. One statement often encodes multiple (counter-)causal relationships and for sub-task 2 the direction is important: while `<e0>A</e0> causes <e1>B</e1>` is causal, `<e1>A</e1> causes <e0>B</e0>` is uncausal.

For the shared task, participants were given access to public training and validation data for developing their systems. The hidden test split was created following the same annotation and construction principles, but was kept private throughout the evaluation. Sub-tasks 1 and 3 were evaluated on all 350 test instances, while sub-task 2 was evaluated only on the ∼900 relationships from the test instances labeled as causal or countercausal.

3.3. Participant Approaches

In 2026, 2 teams participated in this task, submitting 2 runs to sub-task 1, 2 runs to sub-task 2, and 2 runs to sub-task 3. For comparison, we added 1 baseline run to sub-task 1, 1 baseline run to sub-task 2, and 1 baseline run to sub-task 3.

Baselines For the baseline, we fine-tune a RoBERTa model separately for each sub-task. Causality detection is formulated as a binary sequence classification problem. For candidate extraction, RoBERTa is trained as a token-level sequence tagger using BIO-style labels to identify event or concept spans that may participate in a causal relationship. For causality identification, the model is fine-tuned as a ternary classifier over marked event pairs, distinguishing between causal, countercausal, and uncausal relationships. All models are trained using early stopping, with model selection based on F_1-score on the validation set.

Team The Overfitters [51] The team fine-tunes a single DeBERTa-v3-base encoder jointly for all three sub-tasks, using separate lightweight task-specific heads for detection, span extraction, and identification. Their approach is motivated by the assumption that the sub-tasks are structurally dependent: the ability to detect causal information should improve the extraction of relevant spans, while span information should in turn support causality identification. Consequently, the model is trained simultaneously on all three tasks.

Causality detection is formulated as a sequence classification problem and optimized using a classification head with Cross-Entropy loss. Candidate extraction is modeled as a span extraction task using BIO tagging, i.e., token classification. For causality identification, special tokens are added to the tokenizer to mark the candidate events (<e0>, </e0>, <e1>, and </e1>). This allows the encoder to attend directly to their positions within the sentence.

The team's experiments indicate that joint training is useful for learning shared causal representations in the backbone transformer, while DeBERTa's position-sensitive encoding is particularly beneficial for classifying marked span pairs. They identify span extraction as the primary remaining bottleneck, as token-level BIO tagging struggles with exact character-level boundaries, and overlapping or nested event candidates.

Team Shrome [52] For causality detection, the team fine-tunes a BERT-base classifier and combines it with the candidate extraction model from sub-task 2 in a voting setup: a text is classified as containing causal information only if the detection model predicts *yes* and the span extraction model identifies at least one span.

The submission focuses particularly on candidate extraction. For this sub-task, the team uses BILOU tagging [53], an extension of BIO tagging, with CRF decoding [54]. To increase the amount of training data and improve boundary detection, the team applies data augmentation through synonym replacement and random insertion, and additionally incorporates pairs from the original CNCv2 training set. They also continue RoBERTa pre-training on this extended corpus. During inference, BILOU tagging is performed with two separately fine-tuned RoBERTa models and one task-specifically adapted RoBERTa variant. Event spans are then predicted using Viterbi decoding over the CRF outputs.

For causality identification, the team effectively adds four new tokens (<e0>, </e0>, <e1>, and </e1>) to the tokenizer and fine-tunes four separate BERT-base classifiers. The final label is selected by averaging the softmax probabilities produced by the four models. To further expand the training data, the team augments the dataset with 2433 automatically generated countercausal samples.

3.4. Evaluation Setup

Submissions were evaluated separately on each of the three sub-tasks. For sub-task 1, causality detection, we treat the task as a binary classification problem: systems decide whether a given text contains causal information, including both causal and countercausal claims, or whether it is uncausal. We report precision, recall, and F_1-score.

Sub-task 2, candidate extraction, is evaluated as a span extraction task. Since systems output character-level spans rather than class labels, exact boundary matching would be overly strict: a prediction may identify the correct event or concept while differing slightly from the gold annotation in its span boundaries. We therefore use the overlap-based measure for character-level text-alignment by Potthast et al. [55]. For each predicted span, precision is computed as the maximum relative overlap with any gold span; analogously, recall is computed as the maximum relative overlap of each gold span with any predicted span. The reported F_1-score is the harmonic mean of these overlap-based precision and recall values. The evaluator further computes a granularity score that penalizes systems for splitting one gold span into multiple overlapping predictions, as well as a granularity-penalized F_1-score and intersection-over-union. The official ranking for sub-task 2 is based on the granularity-penalized F_1-score.

For sub-task 3, causality identification, systems are given a text and two marked spans, e_0 and e_1, and classify the relationship from e_0 to e_1 as causal, countercausal, or uncausal. We evaluate this ternary classification task using macro-averaged precision, recall, and F_1-score, so that all three relationship classes contribute equally to the final score.

The results for all sub-tasks are shown in Table 4. Overall, the results show that Team The Overfitters achieves the highest effectiveness on causality detection, with the best F_1-score and precision, while Team Shrome obtains the highest recall on this sub-task. However, Team Shrome's detection setup is also the most computationally expensive, since it combines a separately fine-tuned BERT classifier with

Table 4
Effectiveness of the classifiers submitted to each of the causality extraction sub-tasks. The baseline submission is shown in gray.

Team	Detection			Event Extraction			Identification		
	F_1-score	Precision	Recall	F_1-score	Precision	Recall	F_1-score	Precision	Recall
The Overfitters	0.884	0.938	0.836	0.587	0.731	0.584	0.778	0.751	0.818
Shrome	0.869	0.848	0.890	0.728	0.764	0.787	0.817	0.796	0.851
Baseline	0.872	0.872	0.872	0.665	0.749	0.709	0.839	0.827	0.853

the candidate extraction model in a voting setup. For candidate extraction, Team Shrome's model is the most effective by a large margin. Surprisingly this proficiency at sub-task 2 does not translate to higher effectiveness at sub-task 1 despite the voting setup. The effectiveness at sub-task 2 may suggest that BILOU tagging with CRF decoding is better suited to precise event boundary detection than the BIO-tagging schemes used by Team The Overfitters and the baseline. In contrast, causality identification remains dominated by the baseline: neither participating team improves over the baseline F_1-score, precision, or recall on this sub-task.

4. Generalizability of Argument Identification in Context

Argument identification is a fundamental prerequisite for discourse analysis across a range of domains, including political debate, online discussion, and scientific reasoning. A key challenge in contemporary argument identification is that benchmarking is typically conducted on highly specialized datasets, which leads models to learn dataset-specific patterns shaped by the respective dataset's topic bias, argument definition, and labeling scheme, rather than developing abstractions that generalize across contexts [56]. In fact, an argument is not solely defined by features like form or content, but also by its pragmatic function in discourse and its contextualized use therein [57]. Just as humans use context to reliably interpret and identify arguments in discourse, so must machines. Hence, this new task explores whether and how contextual cues can support the automated identification of arguments, investigating the impact of different kinds and amounts of contextual information on developing more generalizable and truly task-aligned systems.

Overview Given a sentence from a dataset along with metadata about its provenance, such as the source text and the dataset's annotation guidelines, the task is to predict whether the sentence is annotated as an argument or not. In this cross-dataset setting, participants must develop robust systems that generalize beyond lexical shortcuts to unseen datasets and exploit rich context information.

4.1. Background

Encoder-based transformer models such as BERT [58], designed for contextualized language representation, have demonstrated state-of-the-art performance on established benchmarks. However, recent research suggests that this performance may be driven by the exploitation of spurious correlations [59], shortcut learning [60] and limited generalization in automated argument identification [56].

4.2. Data Description

On the one hand, we resort to a subset from 10 established, publicly available benchmark datasets, identified as most relevant for argument identification [56]. These are: ABSTRCT [61], ACQUA [62], AEC [63], AFS [64], ARGUMINSCI [65], FINARG [66], IAM [67], PE [68], SCIARK [69] and USELEC [70]. Each consists of 1.7k labeled sentences and is stratified into training, development, and test splits using a 60/20/20 ratio. On the other hand, we provide three novel evaluation-only variants derived from

TACO [71] in order to account for model contamination with these publicly accessible datasets. These include the original TACO annotations, TAPE, in which the same tweets were re-annotated according to the PE guidelines, and TAUS, in which they were re-annotated following the USELEC guidelines. The three variants apply distinct annotation schemes to an identical set of 340 tweets, resulting in 1,020 labeled instances overall. None of the three variants includes training or development annotations. Moreover, the re-annotations were conducted by the three main authors of this task and evaluated using Krippendorff's α. The inter-annotator agreement reached 0.7355α for TAPE and 0.7423α for TAUS. Across all datasets, sentences are labeled as *argument* or *no-argument* according to the respective annotation schemes. Accompanying metadata includes sentence IDs, (where available) generated splits, and (where available) context-relevant information via the original data sources, as well as annotation guidelines and corresponding papers. The entire dataset is publicly available.[3]

4.3. Participant Approaches

In 2026, 4 teams participated in this task, submitting 8 runs in total.

Team Context-Awakens [72] The team approached the task of argument identification as a rule-conditioned inference task rather than a supervised classification problem. The team evaluated GPT-5.2, Mistral-Medium-3.1, and Ministral-8B in their instruction-tuned form, without fine-tuning them on the task dataset. Instead, dataset-specific information was injected into the prompt, including argument definitions, annotation guidelines, and document context extracted from the benchmark metadata. To analyze the contribution of this information, the team introduced a context ladder, a framework that progressively adds context to the prompt: generic instructions only, dataset-specific definitions, annotation guidelines, and finally document context. They also conducted robustness experiments using content-removal and word-order shuffling manipulations and performed a contamination audit to assess benchmark leakage. The submitted run employed a full-context prompting strategy, providing GPT-5.2 with all available task-specific information, including dataset definitions, annotation guidelines, and relevant document context. Moreover, evaluation results indicated that explicitly specifying the annotation criteria helped models apply the intended labeling rules consistently across datasets.

Team Arginvariant [73] The team approached argument identification as a guideline-following problem rather than a supervised classification task. Instead of training models on corpus specific labels, the annotation criteria of each corpus were manually distilled into natural language prompts and applied directly at inference time using five open source, instruction-tuned LLMs: DeepSeek-R1-14B, Gemma-2-27B, Gemma-3-27B, Gemma-4-26B, and Mistral-7B. To improve prompt quality, the authors introduced an iterative error-driven refinement process in which misclassifications shared across multiple models were analyzed and translated into additional rules, constraints, and examples. The refinement procedure was validated on the development data and repeated until no further improvements in macro F_1-score were observed. They submitted three runs that differed only in model assignment. Run 1 used a dataset-specific ensemble strategy, selecting the best-performing model for each corpus based on development-set results. Run 2 employed Gemma-2-27B uniformly across all datasets, while Run 3 used Gemma-3-27B throughout.

Team The-Wildcards [74] The team investigated argument identification as a task that depends on the guidelines and compared sentence-only classification with explicit guideline conditioning. Their experiments covered both fine-tuned and off-the-shelf LLMs, evaluating whether providing annotation guidelines improves cross-domain generalization when the same sentence may receive different labels under different annotation schemes. The authors fine-tuned several open-source models using QLoRA, including Mistral-7B, Llama-3.1-8B, DeepSeek-R1-Qwen-14B, and a set of smaller LLMs, while also evaluating off-the-shelf models such as DeepSeek-V4-Pro and DeepSeek-R1-Qwen-14B. Results showed

[3] https://github.com/TomatenMarc/GAIC-2026

that guideline conditioning consistently improved the strongest mid-sized fine-tuned models, with guideline-conditioned Mistral-7B achieving the best overall development performance. Based on these findings, the team submitted three runs. *solo* used guideline-conditioned Mistral-7B as a single model across all datasets. *hybrid* employed dataset-level expert routing, assigning the best-performing model to each dataset and using DeepSeek-V4-Pro for the out-of-domain Twitter variants. *local* followed the same routing strategy but replaced DeepSeek-V4-Pro with DeepSeek-R1-Qwen-14B to avoid reliance on external APIs.

Team Code-Doctors [75] The team investigated how different forms of domain and guideline information affect cross-domain argument identification. They evaluated five approaches ranging from a Random Forest baseline using frozen MiniLM embeddings to transformer-based classifiers incorporating domain metadata and annotation guidelines. The team compared domain-prefix prompting, retrieval-augmented guideline injection, DeBERTa-based classification, and a RoBERTa model augmented with LLM-generated summaries of dataset annotation guidelines. The best-performing approach used RoBERTa-base fine-tuned on inputs containing both a domain identifier and a concise LLM-generated summary of the corresponding annotation guidelines. To generate these summaries, the authors used Gemini-2.5-Flash to distill each guideline document into a concise description of what qualifies as an argument and what does not. The results showed that compact guideline summaries were more effective than retrieving raw guideline chunks through a RAG pipeline, suggesting that concise and targeted contextual information improves cross-domain generalization.

4.4. Evaluation Setup

To evaluate the systems for their generalizability, the macro F_1-score is measured for each test split, along with the overall average of all these values. The systems are primarily evaluated on the newly created, evaluation-only datasets TACO, TAPE and TAUS. For further insights, evaluation results on the established test splits from the held-out benchmark data are also provided but not used for ranking. This setup addresses the risk of data contamination in LLMs and for participants' potential use of additional datasets during training.

Table 5 shows the condensed leaderboard. Team Context-Awakens achieves the best official result with a Main score of 0.7955 using GPT-5.2 in a full-context prompting setup. The first two Arginvariant runs follow closely, relying on guideline-based prompting with instruction-tuned open-source LLMs, including Gemma and Mistral models. Team The-Wildcards' hybrid run reaches a similarly strong Main score by combining fine-tuned models such as Mistral-7B and Llama-3.1-8B with dataset-specific expert routing. The comparison between Main and Supplementary scores also shows that strong performance on public benchmarks does not always transfer to the new test sets. This gap likely arises because systems tuned to the established benchmark distributions generalize less reliably to unseen annotation settings [56], whereas the strongest approaches explicitly incorporated prompting strategies based on task definitions, annotation guidelines, or contextual information to better adapt to each dataset. For example, The-Wildcards' solo run, which is based on a guideline-conditioned fine-tuned Mistral-7B model, achieved the highest Supplementary score but a substantially lower Main score. Similarly, Code-Doctors obtained strong Supplementary performance with a fine-tuned RoBERTa-based model enriched with LLM-generated guideline summaries but ranked lower on the official evaluation-only datasets. On the other hand, the Context-Awakens and Arginvariant teams performed strongly and consistently on both the Main and Supplementary scores, relying on prompting with task- and context-specific information. In contrast, The-Wildcards ranked in the middle of the leaderboard, using a hybrid approach that combined fine-tuning and prompting.

Table 6 provides the corresponding benchmark-level results used to compute these averages.

Table 5
Condensed leaderboard results for Generalizability of Argument Identification in Context. Main reports the average macro F_1-score across the three new evaluation-only datasets (TACO, TAPE, TAUS) used for the official ranking, while Supplementary reports the average across the 10 publicly available training benchmarks.

Team	Name	Main	Supplementary
context-awakens	Full GAIC Testset	0.7955	0.7308
arginvariant	arginvariant_1	0.7834	0.7899
arginvariant	arinvariant_2	0.7829	0.7533
the-wildcards	hybrid	0.7822	0.8128
the-wildcards	local	0.7693	0.8057
arginvariant	arginvariant_3	0.7561	0.6531
code-doctors	run_1	0.6502	0.8000
the-wildcards	solo	0.5938	0.8148

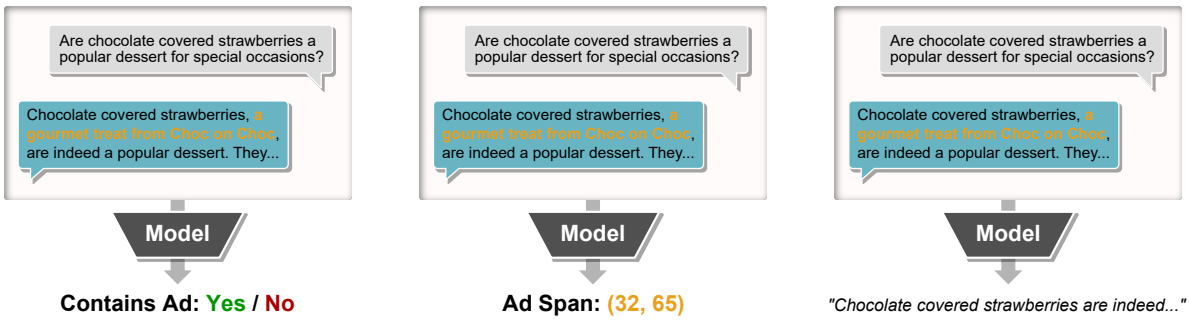
Table 6
Detailed leaderboard results for Generalizability of Argument Identification in Context by benchmark. Each column reports the macro F_1-score for an individual dataset. These scores are aggregated to compute the Main and Supplementary scores reported in Table 5.

Team	Name	ABSTRCT	ACQUA	AEC	AFS	ARGUMINSCI	FINARG	IAM	PE	SCIARK	TACO	TAPE	TAUS	USELEC
context-awakens	Full GAIC Testset	0.8735	0.7882	0.9141	0.7119	0.7933	0.4764	0.7146	0.6646	0.6682	0.8408	0.7604	0.7853	0.7029
arginvariant	arginvariant_1	0.8529	0.8529	0.9588	0.7475	0.8032	0.6705	0.7041	0.8137	0.7465	0.8133	0.7757	0.7612	0.7490
arginvariant	arginvariant_2	0.8466	0.7968	0.9381	0.6441	0.7505	0.6372	0.6795	0.7881	0.7203	0.8133	0.7757	0.7598	0.7316
the-wildcards	hybrid	0.9000	0.8617	0.9588	0.8265	0.8146	0.6728	0.7529	0.7706	0.8234	0.8265	0.7465	0.7735	0.7468
the-wildcards	local	0.9000	0.7910	0.9588	0.8265	0.8146	0.6728	0.7529	0.7706	0.8234	0.7912	0.7506	0.7660	0.7468
arginvariant	arginvariant_3	0.7924	0.7455	0.8076	0.5998	0.6132	0.5598	0.7041	0.4696	0.6760	0.8101	0.7204	0.7377	0.5632
code-doctors	run_1	0.8587	0.8205	0.9559	0.7941	0.8265	0.6715	0.7382	0.7676	0.8056	0.6025	0.6748	0.6734	0.7617
the-wildcards	solo	0.9000	0.8258	0.9588	0.8265	0.8382	0.6728	0.7529	0.7880	0.8234	0.5098	0.6216	0.6499	0.7617

5. Advertisement in Retrieval-Augmented Generation

This task focuses on detecting and subsequently removing native advertising in responses of LLMs and search engines that use retrieval-augmented generation. Both systems play an important role in gathering information on a topic and, consequently, in opinion formation. Given the high costs associated with LLM training and inference [76, 77], commercial vendors have begun adding advertisements to the generated responses as a new source of revenue [78, 79]. While current implementations clearly separate the advertisements from the response text, related work has already proposed different ways of changing the response text in the interest of advertisers, including token sampling from biased distributions [80, 81]. This creates the prospect of LLM responses being biased to influence their users, for example, by presenting the product of an advertiser in a more favorable light than those of the advertiser's competitors.

Overview The task consists of three sub-tasks that build on each other to detect and remove advertising text in LLM responses. In sub-task 1, participants submitted software to detect responses with advertisements. Given a response and a user query, the submissions return a binary label (ad vs. no ad). The goal of sub-task 2 is to locate the advertising text within a response containing advertisements. Based on the response and the user query, the submitted systems return a list of character offsets, indicating the start and end positions of the advertising text. In sub-task 3, these lists of character offsets are used to remove the advertising text while maintaining fluency, factual correctness, and query relevance. Submissions receive the response, the user query, and a list of character offsets, and are asked to output an adapted version of the response. Figure 1 illustrates the sub-tasks.



(a) Sub-Task 1: Detection (b) Sub-Task 2: Span Prediction (c) Sub-Task 3: Blocking

Figure 1: Visualization for the advertisement in retrieval-augmented generation task. In sub-task 1, submissions need to detect if a response contains advertisements or not. In sub-task 2, submissions predict the exact position of advertising text in a response with advertisements. In sub-task 3, submissions block the advertisement by rewriting the response.

5.1. Background

Advertisements inserted into generated responses mimic the text they are placed in, thereby blending in with the surrounding "organic" content. In that regard, they resemble native advertising [82, 83] that mimics the style of articles in newspapers or magazines to conceal its promotional character. Similar to native advertisements, users of LLMs have difficulty detecting advertisements inserted into the generated response text [84, 85]. As users have a high confidence in the information provided by LLMs [86], generated native advertisements may be used to subtly influence their audience.

Previous research has therefore studied the automated detection of these generated advertisements [87], finding transformer-based classifiers to be effective for texts from the same distribution, i.e., generated by the same LLMs and with the same prompts. This was also the case for the submissions to the first iteration of this task [88, 89, 90, 91], which were all highly effective on the test data we made publicly available.[4] The effectiveness dropped, however, for the private test dataset (same as described in Section 5.2), that featured a different distribution of texts, generated by different LLMs and prompts [92]. Many submissions retained high precision scores of > 0.90, but missed a lot of the advertisements with recall scores of $0.35 - 0.50$.

5.2. Data Description

The dataset used for all three sub-tasks is the Webis Generated Native Ads 2025 dataset (WGNA 25).[5] It contains 30 731 responses that were generated by Brave Search, Microsoft Copilot, Perplexity, and YouChat for a total of 9062 queries. Into 13 996 of these responses, we inserted advertisements with LLMs, resulting in a total of 44 727 responses. The products, brands, and services for the advertisements were collected by issuing the 9062 queries to startpage.com and gathering the advertisements for these queries. To increase the stylistic diversity of the generated advertisements, we used a range of different LLMs, specifically: `GPT-4o` and `-mini` via OpenAI, as well as `deepseek-r1`, the 70B parameter versions of `llama-3` and `llama-3.3`, and `qwen-2.5-32b` via groq.[6] Each ad insertion was verified by a set of filters that check if the ad was correctly inserted and the response is otherwise identical to the input. For each response with an ad, the dataset contains the character spans of the advertisements, information on the item that was advertised, and the ID of the original response without advertisements.

Participants used the publicly available version of the WGNA 25 to develop their submissions. For evaluation purposes, we kept a private test set that was created with the same process as outlined above.

[4] https://doi.org/10.5281/zenodo.15270283
[5] https://doi.org/10.5281/zenodo.19736790
[6] https://groq.com/

It was generated using the same RAG-based search engines and LLMs as the public data but features products from different advertisers and responses issued for different queries.

Submissions to sub-task 1 receive all 6748 responses of this private test set alongside the query, the name of search engine that generated the response, and the name of the meta topic of the query like *clothing*, *gaming* or *appliances*. Based on this input, the submissions need to classify the response. For sub-task 2, we limit the input to the 2054 responses with advertisements and otherwise provide the same data as for sub-task 1. Finally, for sub-task 3 we manually selected 100 responses with advertisements, prioritizing cases in which the advertising text was distributed across the response.

5.3. Participant Approaches

In 2026, 3 teams participated in this task, submitting 6 runs to sub-task 1, 2 runs to sub-task 2, and 1 run to sub-task 3. For comparison, we added 1 baseline run to sub-task 1, 2 baseline runs to sub-task 2, and 1 baseline run to sub-task 3.

Baselines For sub-task 1 we fine-tuned a sentence transformer (`MiniLM-L6-v2`) on the WGNA 2025. The model iterates over the response using a window of two sentences, and classifies each pair of sentences as containing an advertisement or not [87]. A response is predicted to contain advertisements if the classifier labels at least one of its sentence pairs as containing an ad. We apply the same sentence transformer to sub-task 2 and use its predictions on the level of sentence pairs to derive the positions of the advertising text in a response. If possible, adjacent sentence pairs are used to narrow down the predictions from sentence pairs to individual sentences. After this step, we return the character offsets of all pairs or sentences predicted to contain advertising text. As a naive baseline for sub-task 2, we added an approach that always predicts the last two sentences of a response to contain advertisements. This is motivated from the observation that the LLMs mentioned in Section 5.2 often added advertisements after the rest of the response. Each response is split into sentences using regular expressions and we return the character offsets of the last two sentences as the prediction. For sub-task 3, we added a naive baseline that removes all advertising text without replacement.

Team Code-Doctors [75] As a baseline model for sub-task 1, the team submitted a Random Forest classifier trained on embeddings created by `all-MiniLM-L6-v2` for the concatenated query and response. The main approach is a fine-tuned RoBERTa base model, again based on concatenated queries and responses, and adding a square-root class weight to the cross-entropy loss function to account for the class imbalance of responses with and without ads. For sub-task 2, the team also fine-tuned a RoBERTa model, this time on sentence pairs. To account again for the class imbalance, an inverse-frequency class weighting is applied. The model predicts three classes—no ad, ad in sentence 1, or ad in sentence 2. Sentences with ads are then located in the full text by exact string matching. For sub-task 3, the team tested different instruction-tuned LLMs to rewrite responses with ads, but observed difficulties like tense inconsistencies or added content with this approach. For this reason, they only submitted a rule-based approach, where advertising passages are deleted, and the resulting text is smoothed by removing blank lines and repeated spaces.

Team Modernbert-Adhunter This team participated in sub-tasks 1 and 2 with a single multi-task model, fine-tuning `ModernBERT-base` end-to-end. A shared encoder feeds two linear heads: a document-level classifier on the `[CLS]` representation and a token-level classifier over all token states. Both heads were trained jointly by summing their cross-entropy losses, with the tokenized query and response as input, and tokens labeled in a binary inside/outside scheme. For sub-task 1, the document label is not taken from the document head, but derived from the span extractor: a response is labeled as advertising if at least one advertising span is found. To find advertising spans (sub-task 2), long responses are processed with a sliding window (512 tokens); per-token ad probabilities are averaged across overlapping windows, and a threshold tuned on the validation set is applied for assigning the

Table 7
Effectiveness of the classifiers submitted to sub-task 1. The submissions are ranked by their F_1-score on the task of classifying whether a response contains an advertisement or not. The Baseline submission ("minilm") is shown in gray.

Rank	Team	Approach	Precision	Recall	F_1-score
1	ZHAW	CompactQueryClassifier	0.990	0.981	0.985
2	Baseline	minilm	0.930	0.900	0.915
3	ZHAW	QwenResidualStackV2	0.608	0.671	0.638
4	ZHAW	QwenResidualOnlyV2	0.616	0.610	0.613
5	ZHAW	BaseStack	0.532	0.691	0.601
6	ModernBERT-AdHunter	jovial-branch	0.942	0.293	0.447
7	Code-Doctors	RoBERTa fine-tuned	0.985	0.218	0.357

Table 8
Effectiveness of the classifiers submitted to sub-task 1 on the test split of the Webis Generated Native Ads 2024 dataset. The submissions are ranked by their F_1-score on the task of classifying whether a response contains an advertisement or not. The Baseline submission ("minilm") is shown in gray. The BaseStack submission by team ZHAW failed to run on the 2024 dataset.

Rank	Team	Approach	Precision	Recall	F_1-score
1	ZHAW	CompactQueryClassifier	0.918	0.826	0.876
2	Code-Doctors	RoBERTa fine-tuned	0.782	0.731	0.755
3	ModernBERT-AdHunter	jovial-branch	0.609	0.828	0.702
4	Baseline	minilm	0.873	0.406	0.555
5	ZHAW	QwenResidualStackV2	0.472	0.426	0.448
6	ZHAW	QwenResidualOnlyV2	0.460	0.437	0.448
—	ZHAW	BaseStack	—	—	—

advertising label. The resulting advertising labels are merged and finally split into sentence-level spans by a punctuation heuristic.

Team ZHAW [93] The team participated only in sub-task 1, submitting four different configurations. Three use `all-mpnet-base-v2` embeddings classified by logistic regression: a response-only baseline, and two variants that add a neutral, brand-free reference text generated by `Qwen2.5-1.5B-Instruct`, using either the concatenation of the response embedding, the reference embedding, and their difference or the difference vector alone. The fourth and best configuration consists of an ALBERT model, fine-tuned end-to-end on the concatenated query and response. This submission reaches a test F_1 of 0.985 versus 0.60–0.64 for the embedding-based setups. The Qwen neutralisation residual gives only a small improvement over the plain response embedding (F_1 0.638 vs. 0.601), well below the fine-tuned ALBERT model.

5.4. Evaluation Setup

For sub-task 1 we use conventional classification metrics, specifically precision, recall, F_1-score, and accuracy. The final ranking is done based on the submission's F_1-score on the private test set mentioned in Section 5.2.

The evaluation for sub-task 2 is also based on precision, recall, and F_1-score. For this task, however, the measures are calculated for each response based on the overlap between the ground truth character offsets and the predictions for that response. The final score of a submission is the mean of the

Table 9
Effectiveness of the classifiers submitted to sub-task 2. The submissions are ranked by their F_1-score on the task of predicting the spans of advertising text in a response with advertisements. Baseline submissions ("minilm" and "last-sentences") are shown in gray.

Rank	Team	Approach	Precision	Recall	F_1-score
1	Baseline	minilm	0.821	0.657	0.697
2	Code-Doctors	RoBERTa sentence-pair (3-class)	0.726	0.719	0.668
3	Baseline	last-sentences	0.605	0.454	0.472
4	ModernBERT-AdHunter	messy-key	0.279	0.178	0.198

Table 10
Effectiveness of the software submitted to sub-task 3. Since the submission by team Code-Doctors and the Baseline ("remove", in gray) are identical, the annotations were reused. The blocked responses were evaluated manually for each of the three metrics on a four-point scale from 0 to 3.

Rank	Team	Approach	Correctness	Fluency	Relevance
1	Code-Doctors	Rule-based span removal	2.763	2.180	2.883
2	Baseline	remove	2.763	2.180	2.883

response-level scores. In addition to that, we calculate a "granularity"-adjusted F_1-score, inspired by the PlagDet-score used in a similar shared task [55]. The goal of this adjustment is to punish repeated detection of the same ground truth span of advertising text. The submissions were ranked according to their F_1-score on the 2054 responses of the private test set that did not contain advertisements. The granularity-adjustment was not necessary to distinguish between the submissions.

For sub-task 3, three annotators manually evaluated the responses after blocking the advertisements. Additionally, we experimented with LLM-as-a-judge with deepeval and `gpt-5.4-mini`.[7] However, since there was virtually no agreement between the LLM and any of the annotators (Kendall's τ of around 0), we discarded its judgments for the evaluation. The annotators evaluated each response on a four-point scale from 0 to 3 for correctness, fluency, and query relevance. The correctness was assessed by using the original response without advertisements from the WGNA 25 as reference to ensure that no new statements were added in the process of blocking. In addition to the blocked response and the original response, the annotators received the query and the input response with advertisements. The final scores of a submission are the means across all responses. To measure the annotator agreement, we used Gwet's AC2 [94], because the annotation distributions were highly skewed towards the category 3, and alternatives like Krippendorff's α or Kendall's τ are not robust to these imbalances. Across the three annotators, we observed a very high agreement for relevance (0.94) and correctness (0.86), and a high agreement for fluency (0.70).

The results for the sub-tasks are summarized in Tables 7-10.

Sub-Task 1 The most effective submission for sub-task 1 was the fine-tuned ALBERT model by team ZHAW, achieving an F_1-score of 0.985. This is followed by the MiniLM baseline with an F_1 of 0.915. The embedding-based logistic regression models by team ZHAW achieve F_1-scores of 0.638 (QwenResidualStackV2), 0.613 (QwenResidualOnlyV2), and 0.601 (BaseStack). The submission by ModernBERT-AdHunter achieves an F_1-score of 0.447. The model outperforms the MiniLM baselines on precision (0.942), but misses a lot of the responses with advertisements, resulting in a recall of 0.293. Similarly, the submission by team Code-Doctors achieves a very high precision of 0.985, but a low recall of 0.218, resulting in an F_1-score of 0.357.

In addition to the private test set of the WGNA 25 dataset, we evaluated the submissions on the test set

[7] deepeval.com

of the predecessor dataset WGNA 24.[8] The results are summarized in Table 8. Team ZHAW's ALBERT model is also the most effective submission on this data. Despite losing some of its effectiveness on this out-of-distribution data, it still achieves a high F_1-score of 0.876. The submission by team Code-Doctors loses some of its precision, but gains a lot of recall, resulting in an overall increase in F_1-score to 0.755. The same is true for the submission by team ModernBERT-AdHunter that increases its F_1-score to 0.702. A possible explanation for the similarities between these two submissions could be that both set a prediction threshold that is too strict for the private test dataset. The advertisements in the WGNA 24 dataset, on the other hand, appear to be easier to detect for these submissions, leading to an overall improvement in effectiveness at the expense of precision. The other submissions are not robust to the distribution shift. The MiniLM baseline in particular drops noticeably to an F_1-score of 0.555. The QwenResidual-submissions by team ZHAW achieve F_1-scores of 0.448, and their BaseStack submission fails to process this dataset.

Sub-Task 2 On sub-task 2, the MiniLM baseline achieves an F_1-score of 0.697, followed by the submission by team Code-Doctors with an F_1-score of 0.668. The naive baseline of always predicting the last two sentences achieves an F_1-score of 0.472 and a decent precision of 0.605, indicating that the LLMs often placed their advertisements at the end of the response. ModernBERT-AdHunter submitted the same model to sub-tasks 1 and 2. For sub-task 2 it achieved a low F_1-score of 0.198. In contrast to its high precision for sub-task 1, the model only achieved a precision of 0.279, suggesting that exactly locating the advertising text is more challenging than the binary classification on response-level.

Sub-Task 3 For sub-task 3, the only submission was by team Code-Doctors. As the team found LLM-based rewriting to produce unwanted artifacts like added content, they submitted a simple rule-based approach that removes the input spans without replacement and cleans up potential whitespace issues. Apart from the whitespace cleanup, this is identical to our baseline and we used the same annotations for both. As no new content was added, both relevance and correctness were evaluated highly at an average of 2.883 and 2.763, respectively. The removal without replacement, however, negatively affected response fluency. While effective for full-sentence advertisements, this approach produced some interrupted sentences that led to low scores by the annotators.

6. Conclusion

The seventh edition of the Touché lab on argumentation systems featured four tasks.

In Fallacy Detection (new task), fine-tuned encoder models emerged as the dominant choice across all three sub-tasks. Binary fallacy detection (sub-task 1) drew the widest range of methods, from small-model ensembles and classical TF-IDF stacks to multi-agent LLM pipelines, whereas fallacy classification (sub-task 2) and scheme classification (sub-task 3) were tackled almost exclusively with fine-tuned encoders. In Causality Extraction (new task), participants relied on encoder-only transformer models across the sub-tasks. Both teams used variants of BIO-style tagging for sub-task 2, but their overall strategies differed substantially: one team trained a shared backbone jointly across all tasks, whereas the other used multiple task-specific model variants and combined their predictions through ensemble averaging. In Generalizability of Argument Identification in Context (new task), participants largely framed argument identification as a guideline-following problem rather than a conventional supervised classification task. Approaches ranged from prompt-based inference with instruction-tuned LLMs to fine-tuned transformer models augmented with annotation guidelines and domain metadata. Across teams, a common finding was that explicitly incorporating annotation criteria, either through prompting, guideline summaries, or guideline-conditioned fine-tuning, improved cross-domain performance and helped models adapt to dataset-specific definitions of arguments. In Advertisement in Retrieval-Augmented Generation (2nd edition), participants mainly relied on fine-tuning different transformer models for the two ad detection tasks (document and span level). The third task aiming at

[8] https://doi.org/10.5281/zenodo.15270283

the removal of advertising passages, was addressed by one team with a simple rule-based approach, which was found to outperform LLM-based rewriting.

We plan to continue Touché as a collaborative platform for researchers in argumentation systems. All Touché resources are freely available, including topics, manual relevance, argument quality, and stance judgments, and submitted runs from participating teams. In all Touché labs combined, we received 460 runs from 126 teams. We manually labeled the relevance and quality of more than 36,000 argumentative texts, web documents, and images.[9] These resources and other events such as workshops will help to further foster the community working on argumentation systems.

7. Acknowledgments

This work was partially supported by the European Commission under grant agreement GA 101070014 (https://openwebsearch.eu) and by the German Federal Ministry of Research, Technology, and Space (BMFTR) through the project "DIALOKIA: Überprüfung von LLM-generierter Argumentation mittels dialektischem Sprachmodell" (01IS24084A-B).

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] 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.

[3] T. Hagen, M. Fröbe, J. H. Merker, H. Scells, M. Hagen, M. Potthast, TIREx Tracker: The Information Retrieval Experiment Tracker, in: 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2025), ACM, 2025. doi:10.1145/3726302.3730297.

[4] C. L. Hamblin, Fallacies, Methuen & Co. Ltd, London, 1970. SBN 416 14570 1.

[5] D. N. Walton, C. Reed, F. Macagno, Argumentation Schemes, Cambridge University Press, 2008.

[6] S. Sahai, O. Balalau, R. Horincar, Breaking Down the Invisible Wall of Informal Fallacies in Online Discussions, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), 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, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, Association for Computational Linguistics, 2021, pp. 644–657. doi:10.18653/v1/2021.acl-long.53.

[7] F. Macagno, A means-end classification of argumentation schemes, in: F. H. van Eemeren, B. Garssen (Eds.), Reflections on Theoretical Issues in Argumentation Theory, Springer International Publishing, Cham, 2015, pp. 183–201. doi:10.1007/978-3-319-21103-9_14.

[8] F. Macagno, Argumentation profiles and the manipulation of common ground. The arguments of populist leaders on Twitter, Journal of Pragmatics 191 (2022) 67–82. doi:10.1016/j.pragma.2022.01.022.

[9] Z. Sourati, F. Ilievski, H. Sandlin, A. Mermoud, Case-Based Reasoning with Language Models for Classification of Logical Fallacies, in: Proceedings of the Thirty-Second International Joint

[9] Judgments are publicly available at the lab's web page, https://touche.webis.de

Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China, ijcai.org, 2023, pp. 5188–5196. doi:10.24963/ijcai.2023/576.

[10] J. Jeong, H. Jang, H. Park, Large Language Models Are Better Logical Fallacy Reasoners with Counterargument, Explanation, and Goal-Aware Prompt Formulation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025, Association for Computational Linguistics, 2025, pp. 6918–6937. doi:10.18653/v1/2025.findings-naacl.384.

[11] P. Goffredo, S. Haddadan, V. Vorakitphan, E. Cabrio, S. Villata, Fallacious Argument Classification in Political Debates, in: L. D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, ijcai.org, 2022, pp. 4143–4149. doi:10.24963/ijcai.2022/575.

[12] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical Fallacy Detection, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, Association for Computational Linguistics, 2022, pp. 7180–7198. doi:10.18653/V1/2022.FINDINGS-EMNLP.532.

[13] A. Lalwani, T. Kim, L. Chopra, C. Hahn, Z. Jin, M. Sachan, Autoformalizing Natural Language to First-Order Logic: A Case Study in Logical Fallacy Detection, 2025. doi:10.48550/arxiv.2405.02318. arXiv:2405.02318.

[14] G. D. S. Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, P. Nakov, Fine-Grained Analysis of Propaganda in News Article, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, Association for Computational Linguistics, 2019, pp. 5635–5645. doi:10.18653/v1/d19-1565.

[15] G. D. S. Martino, A. Barrón-Cedeño, H. Wachsmuth, R. Petrov, P. Nakov, SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles, in: A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, E. Shutova (Eds.), Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING 2020, Barcelona (online), December 12-13, 2020, International Committee for Computational Linguistics, 2020, pp. 1377–1414. doi:10.18653/v1/2020.semeval-1.186.

[16] R. Ruiz-Dolz, J. Lawrence, Detecting Argumentative Fallacies in the Wild: Problems and Limitations of Large Language Models, in: M. Alshomary, C. Chen, S. Muresan, J. Park, J. Romberg (Eds.), Proceedings of the 10th Workshop on Argument Mining, ArgMining 2023, Singapore, December 7, 2023, Association for Computational Linguistics, 2023, pp. 1–10. doi:10.18653/v1/2023.argmining-1.1.

[17] M. Glockner, Y. Hou, P. Nakov, I. Gurevych, Missci: Reconstructing Fallacies in Misrepresented Science, in: L. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, Association for Computational Linguistics, 2024, pp. 4372–4405. doi:10.18653/v1/2024.acl-long.240.

[18] S. Simhadri, ArguMind at Touché: A Unified Multi-Task Transformer Approach for Fallacy and Argument Scheme Classification, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[19] A. Ghiriti, Graz University of Technology at Touché: Prompting Strategies for LLM-based Fallacy Detection, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[20] D. Markova, helium at Touch'e: The Case for Naturalistic Evaluation of Fallacy Detection, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[21] K. Alexander, M. Z. Ullah, D. Gkatzia, NapierNLP at Touché: Fallacy Detection with SLMs and Model Fusion, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[22] S. Singh, M. Kumar, A. Jamatia, NIT-Agartala-NLP Team at Touché: Detecting and Classifying

Fallacies in Reddit Arguments Using Ensemble Methods, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[23] R. Lin, Y. Yi, OoOoO at Touché: Argument-Structure-Aware RoBERTa Ensembles for Fallacy Detection, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[24] M. Smith, M. Islam, C. Seals, Seals-NLP at Touché: Comparing Fully Tuned Encoders and Optimized LLMs for Multi-Task Fallacy Detection, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[25] İrem Batıgün, Y. Öztemir, Team Sophisma at Touché: Transformer-Based Fallacy Detection with Enhanced Argument Representations, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[26] S. Psychias, UZHCL_for_the_win at Touché: Encoder Fine-Tuning, Synthetic Augmentation, and an LLM Detour, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[27] F.-Z. Rezkellah, S. Conrad, uzh_tt at Touché: Confidence-Adaptive Cascaded Fallacy Detection, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[28] F. B. Memar, J. de Rijke, M. den Ouden, N. Biesterbos, Veritas at Touché: Fine-tuning Encoder Models for Fallacy Detection and Classification, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[29] C. Mathews, M. Heinrich, B. Stein, Webis at Touché: The Court of Fallacies: Adversarial LLM Advocacy with Forensic Appeal for Reddit Fallacy Detection, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[30] A. Bondarenko, M. Wolska, S. Heindorf, L. Blübaum, A.-C. N. Ngomo, B. Stein, P. Braslavski, M. Hagen, M. Potthast, CausalQA: A Benchmark for Causal Question Answering, in: N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, S.-H. Na (Eds.), Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, International Committee on Computational Linguistics, 2022, pp. 3296–3308.

[31] R. Ceraolo, D. Kharlapenko, A. Khan, A. Reymond, R. Mihalcea, B. Schölkopf, M. Sachan, Z. Jin, Quriosity: Analyzing Human Questioning Behavior and Causal Inquiry through Curiosity-Driven Queries, in: K. Inui, S. Sakti, H. Wang, D. F. Wong, P. Bhattacharyya, B. Banerjee, A. Ekbal, T. Chakraborty, D. P. Singh (Eds.), Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP-AACL 2025, Mumbai, India, December 20-24, 2025, The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, 2025, pp. 534–563.

[32] F. A. Tan, H. Hettiarachchi, A. Hürriyetoglu, N. Oostdijk, T. Caselli, T. Nomoto, O. Uca, F. F. Liza, S.-K. Ng, RECESS: Resource for Extracting Cause, Effect, and Signal Spans, in: J. C. Park, Y. Arase, B. Hu, W. Lu, D. Wijaya, A. Purwarianti, A. A. Krisnadhi (Eds.), 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 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023, Association for Computational Linguistics, 2023, pp. 66–82. doi:10.18653/V1/2023.IJCNLP-MAIN.6.

[33] R. Girju, Automatic Detection of Causal Relations for Question Answering, in: Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering, Association for Computational Linguistics, Sapporo, Japan, 2003, pp. 76–83. doi:10.3115/1119312.1119322.

[34] Q.-C. Bui, B. Ó. Nualláin, C. A. Boucher, P. M. A. Sloot, Extracting Causal Relations on HIV Drug Resistance from Literature, BMC Bioinform. 11 (2010) 101. doi:10.1186/1471-2105-11-101.

[35] Y. Cao, P. Zhang, J. Guo, L. Guo, Mining Large-scale Event Knowledge from Web Text, in: D. Abramson, M. Lees, V. V. Krzhizhanovskaya, J. J. Dongarra, P. M. A. Sloot (Eds.), Proceedings of the International Conference on Computational Science, ICCS 2014, Cairns, Queensland, Australia, 10-12 June, 2014, volume 29 of *Procedia Computer Science*, Elsevier, 2014, pp. 478–487. doi:10.1016/J.PROCS.2014.05.043.

[36] P. Mirza, Extracting Temporal and Causal Relations between Events, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Student Research Workshop, The Association for Computer Linguistics, 2014, pp. 10–17. doi:10.3115/V1/P14-3002.

[37] P. Li, K. Mao, Knowledge-Oriented Convolutional Neural Network for Causal Relation Extraction from Natural Language Texts, Expert Syst. Appl. 115 (2019) 512–523. doi:10.1016/J.ESWA.2018.08.009.

[38] Z. Li, Q. Li, X. Zou, J. Ren, Causality Extraction Based on Self-Attentive BiLSTM-CRF with Transferred Embeddings, Neurocomputing 423 (2021) 207–219. doi:10.1016/J.NEUCOM.2020.08.078.

[39] V. Khetan, R. R. Ramnani, M. Anand, S. Sengupta, A. E. Fano, Causal BERT: Language Models for Causality Detection Between Events Expressed in Text, in: K. Arai (Ed.), Intelligent Computing - Proceedings of the 2021 Computing Conference, Volume 1, SAI 2021, Virtual Event, 15-16 July, 2021, volume 283 of *Lecture Notes in Networks and Systems*, Springer, 2021, pp. 965–980. doi:10.1007/978-3-030-80119-9_64.

[40] P. Hosseini, D. A. Broniatowski, M. T. Diab, Predicting Directionality in Causal Relations in Text, CoRR abs/2103.13606 (2021). arXiv:2103.13606.

[41] J. Yang, H. Xiong, H. Zhang, M. Hu, N. An, Causal Pattern Representation Learning for Extracting Causality from Literature, in: Proceedings of the 5th International Conference on Machine Learning and Natural Language Processing, MLNLP 2022, Sanya, China, December 23-25, 2022, ACM, 2022, pp. 229–233. doi:10.1145/3578741.3578787.

[42] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. K. Joshi, B. L. Webber, The Penn Discourse TreeBank 2.0, in: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco, European Language Resources Association, 2008.

[43] C. Hidey, K. McKeown, Identifying Causal Relations Using Parallel Wikipedia Articles, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers, The Association for Computer Linguistics, 2016. doi:10.18653/V1/P16-1135.

[44] J. Yang, S. C. Han, J. Poon, A Survey on Extraction of Causal Relations from Natural Language Text, Knowledge and Information Systems 64 (2022) 1161–1186. doi:10.1007/s10115-022-01665-w.

[45] C. Mihaila, T. Ohta, S. Pyysalo, S. Ananiadou, BioCause: Annotating and Analysing Causality in the Biomedical Domain, BMC Bioinform. 14 (2013) 2. doi:10.1186/1471-2105-14-2.

[46] J. Dunietz, L. S. Levin, J. G. Carbonell, The BECauSE Corpus 2.0: Annotating Causality and Overlapping Relations, in: N. Schneider, N. Xue (Eds.), Proceedings of the 11th Linguistic Annotation Workshop, LAW@EACL 2017, Valencia, Spain, April 3, 2017, Association for Computational Linguistics, 2017, pp. 95–104. doi:10.18653/V1/W17-0812.

[47] D. Mariko, H. Abi-Akl, E. Labidurie, S. Durfort, H. De Mazancourt, M. El-Haj, The Financial Document Causality Detection Shared Task (FinCausal 2020), in: D. M. El-Haj, D. V. Athanasakou, D. S. Ferradans, D. C. Salzedo, D. A. Elhag, D. H. Bouamor, D. M. Litvak, D. P. Rayson, D. G. Giannakopoulos, N. Pittaras (Eds.), Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, COLING, Barcelona, Spain (Online), 2020, pp. 23–32.

[48] V. Khetan, S. Wadhwa, B. C. Wallace, S. Amir, SemEval-2023 Task 8: Causal Medical Claim Identification and Related PIO Frame Extraction from Social Media Posts, in: A. K. Ojha, A. S. Dogruöz, G. D. S. Martino, H. T. Madabushi, R. Kumar, E. Sartori (Eds.), Proceedings of the The 17th International Workshop on Semantic Evaluation, SemEval@ACL 2023, Toronto, Canada,

13-14 July 2023, Association for Computational Linguistics, 2023, pp. 2266–2274. doi:10.18653/v1/2023.SEMEVAL-1.311.

[49] P. G. Corral, H. Béchara, R. Zhang, S. Jankin, PolitiCause: An Annotation Scheme and Corpus for Causality in Political Texts, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy, ELRA and ICCL, 2024, pp. 12836–12845.

[50] T. Hagen, N. Deckers, F. Wolter, H. Scells, M. Potthast, Investigating Counterclaims in Causality Extraction from Text, CoRR abs/2510.08224 (2025).

[51] L. Ruttert, M. Gü"lhan, The Overfitters at Touché: Joint Causality Extraction with DeBERTa-v3, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[52] R. Z. Nobari, S. Sooratgar, Shrome at Touché: Soft-Vote Ensembling and Counter-Causal Augmentation for Causality Extraction, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[53] 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: A. Hürriyetoğlu, H. Tanev, V. Zavarella, R. Yeniterzi, E. Yörük, M. Slavcheva (Eds.), Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text, INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria, 2023, pp. 38–43. URL: https://aclanthology.org/2023.case-1.5/.

[54] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in: C. E. Brodley, A. P. Danyluk (Eds.), Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001, Morgan Kaufmann, 2001, pp. 282–289.

[55] M. Potthast, T. Gollub, M. Hagen, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos, B. Stein, Overview of the 5th International Competition on Plagiarism Detection, in: P. Forner, R. Navigli, D. Tufis (Eds.), Working Notes Papers of the CLEF 2013 Evaluation Labs, volume 1179 of *Lecture Notes in Computer Science*, 2013. URL: https://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-PotthastEt2013.pdf.

[56] M. Feger, K. Boland, S. Dietze, Limited Generalizability in Argument Mining: State-Of-The-Art Models Learn Datasets, Not Arguments, 2025. URL: https://arxiv.org/abs/2505.22137. arXiv:2505.22137.

[57] F. H. van Eemeren, B. Garssen, E. C. W. Krabbe, A. F. Snoeck Henkemans, B. Verheij, J. H. M. Wagemans, Handbook of Argumentation Theory, 1 ed., Springer, Dordrecht, 2014. doi:10.1007/978-90-481-9473-5, 61 b/w illustrations, 18 colour illustrations.

[58] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[59] T. S. Thorn Jakobsen, M. Barrett, A. Søgaard, Spurious Correlations in Cross-Topic Argument Mining, in: L.-W. Ku, V. Nastase, I. Vulić (Eds.), Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, Online, 2021, pp. 263–277. doi:10.18653/v1/2021.starsem-1.25.

[60] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F. A. Wichmann, Shortcut learning in deep neural networks, Nature Machine Intelligence 2 (2020) 665–673. doi:10.1038/s42256-020-00257-z.

[61] T. Mayer, E. Cabrio, S. Villata, Transformer-Based Argument Mining for Healthcare Applications, in: European Conference on Artificial Intelligence, 2020.

[62] A. Panchenko, A. Bondarenko, M. Franzek, M. Hagen, C. Biemann, Categorizing Comparative Sentences, in: B. Stein, H. Wachsmuth (Eds.), Proceedings of the 6th Workshop on Argument Mining, Association for Computational Linguistics, Florence, Italy, 2019, pp. 136–145. doi:10.

18653/v1/W19-4516.

[63] R. Swanson, B. Ecker, M. Walker, Argument Mining: Extracting Arguments from Online Dialogue, in: A. Koller, G. Skantze, F. Jurcicek, M. Araki, C. P. Rose (Eds.), Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics, Prague, Czech Republic, 2015, pp. 217–226. doi:10.18653/v1/W15-4631.

[64] A. Misra, B. Ecker, M. Walker, Measuring the Similarity of Sentential Arguments in Dialogue, in: R. Fernandez, W. Minker, G. Carenini, R. Higashinaka, R. Artstein, A. Gainer (Eds.), Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics, Los Angeles, 2016, pp. 276–287. doi:10.18653/v1/W16-3636.

[65] A. Lauscher, G. Glavaš, S. P. Ponzetto, An Argument-Annotated Corpus of Scientific Publications, in: N. Slonim, R. Aharonov (Eds.), Proceedings of the 5th Workshop on Argument Mining, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 40–46. doi:10.18653/v1/W18-5206.

[66] A. Alhamzeh, R. Fonck, E. Versmée, E. Egyed-Zsigmond, H. Kosch, L. Brunie, It's Time to Reason: Annotating Argumentation Structures in Financial Earnings Calls: The FinArg Dataset, in: C.-C. Chen, H.-H. Huang, H. Takamura, H.-H. Chen (Eds.), Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP), Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 2022, pp. 163–169. doi:10.18653/v1/2022.finnlp-1.22.

[67] L. Cheng, L. Bing, R. He, Q. Yu, Y. Zhang, L. Si, IAM: A Comprehensive and Large-Scale Dataset for Integrated Argument Mining Tasks, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2277–2287. doi:10.18653/v1/2022.acl-long.162.

[68] C. Stab, I. Gurevych, Parsing Argumentation Structures in Persuasive Essays, Computational Linguistics 43 (2017) 619–659. doi:10.1162/COLI_a_00295.

[69] A. Fergadis, D. Pappas, A. Karamolegkou, H. Papageorgiou, Argumentation Mining in Scientific Literature for Sustainable Development, in: K. Al-Khatib, Y. Hou, M. Stede (Eds.), Proceedings of the 8th Workshop on Argument Mining, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 100–111. doi:10.18653/v1/2021.argmining-1.10.

[70] S. Haddadan, E. Cabrio, S. Villata, Yes, We Can! Mining Arguments in 50 Years of US Presidential Campaign Debates, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4684–4690. doi:10.18653/v1/P19-1463.

[71] M. Feger, S. Dietze, TACO – Twitter Arguments from COnversations, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 15522–15529. URL: https://aclanthology.org/2024.lrec-main.1349/.

[72] J. Widera, Context-Awakens at Touché: Generalizable Argument Identification with In-Context Learning, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[73] M. Kanadan, M. Heinrich, B. Stein, arginvariant at Touché: Error-Driven Prompt Refinement for Argument Identification, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[74] M. Q. Hussain, M. Heinrich, M. Kandadan, B. Stein, The Wildcards at Touché: Guideline-Conditioned Argument Identification with Dataset-Level Expert Routing, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[75] A. Siddiqui, M. Alam, Z. Aslam, F. Alvi, A. Samad, Code-Doctors at Touché: Cross-Domain Argument Detection and Advertisement Removal in RAG Responses, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[76] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large Language

Models: A Survey, 2024. doi:10.48550/ARXIV.2402.06196, version Number: 3.
[77] G. Varoquaux, S. Luccioni, M. Whittaker, Hype, Sustainability, and the Price of the Bigger-is-Better Paradigm in AI, in: Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency, FAccT '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 61–75. doi:10.1145/3715275.3732006.
[78] OpenAI, Our Approach to Advertising and Expanding Access to ChatGPT, 2026. URL: https://openai.com/index/our-approach-to-advertising-and-expanding-access/.
[79] Perplexity, Why We're Experimenting with Advertising, 2024. URL: https://www.perplexity.ai/hub/blog/why-we-re-experimenting-with-advertising.
[80] P. Dütting, V. Mirrokni, R. Paes Leme, H. Xu, S. Zuo, Mechanism Design for Large Language Models, in: Proceedings of the ACM Web Conference 2024, ACM, Singapore Singapore, 2024, pp. 144–155. doi:10.1145/3589334.3645511.
[81] E. Soumalias, M. J. Curry, S. Seuken, Truthful Aggregation of LLMs with an Application to Online Advertising, 2024. doi:10.48550/ARXIV.2405.05905, version Number: 5.
[82] E. E. Schauster, P. Ferrucci, M. S. Neill, Native Advertising is the New Journalism: How Deception Affects Social Responsibility, American Behavioral Scientist 60 (2016) 1408–1424.
[83] B. W. Wojdynski, N. J. Evans, Going native: Effects of disclosure position and language onthe recognition and evaluation of online native advertising, Journal of Advertising 45 (2016) 157–168.
[84] I. Zelch, M. Hagen, M. Potthast, A User Study on the Acceptance of Native Advertising in Generative IR, in: ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR 2024), ACM, 2024. doi:10.1145/3627508.3638316.
[85] B. J. Tang, K. Sun, N. T. Curran, F. Schaub, K. G. Shin, Ads that Talk Back: Implications and Perceptions of Injecting Personalized Advertising into LLM Chatbots, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 9 (2025) 1–47. doi:10.1145/3770640.
[86] S. E. Spatharioti, D. M. Rothschild, D. G. Goldstein, J. M. Hofman, Comparing Traditional and LLM-based Search for Consumer Choice: A Randomized Experiment, CoRR abs/2307.03744 (2023). doi:10.48550/ARXIV.2307.03744.
[87] S. Schmidt, I. Zelch, J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Detecting Generated Native Ads in Conversational Search, in: Companion Proceedings of the ACM Web Conference 2024, WWW '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 722–725. doi:10.1145/3589335.3651489.
[88] T. A. Bouhairi, A. Alhamzeh, Pirate Passau at Touché: Do We Need to Get Complex? A Comparative Analysis of Traditional and Advanced NLP Approaches for Advertisement Classification, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 4563–4570. URL: https://ceur-ws.org/Vol-4038/paper_379.pdf.
[89] A. Dutta, A. Majumdar, S. Biswas, D. Das, S. Bandyopadhyay, JU-NLP at Touché: Covert Advertisement in Conversational AI-Generation and Detection Strategies, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 4586–4600. URL: https://ceur-ws.org/Vol-4038/paper_382.pdf.
[90] S. Kamani, M. Taqi, M. A. Chaudhary, M. A. H. Hanif, F. Alvi, A. Samad, Git Gud at Touché: Unified RAG Pipeline for Native Ad Generation and Detection, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_384.pdf.
[91] T. E. Kim, J. Coelho, G. Onilude, J. Singh, TeamCMU at Touché: Adversarial Co-Evolution for Advertisement Integration and Detection in Conversational Search, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 4619–4634. URL: https://ceur-ws.org/Vol-4038/paper_385.pdf.
[92] J. Kiesel, Ç. Çöltekin, M. Gohsen, S. Heineking, M. Heinrich, M. Fröbe, T. Hagen, M. Aliannejadi, S. Anand, T. Erjavec, M. Hagen, M. Kopp, N. Ljubešić, K. Meden, N. Mirzakhmedova, V. Morkevičius, H. Scells, M. Wolter, I. Zelch, M. Potthast, B. Stein, Overview of Touché 2025: Argumentation Systems, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. García Seco de Herrera, J. Mothe, F. Piroi,

P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. 16th International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2025.

[93] S. Stalder, ZHAW at Touché: Advertisement Detection in RAG Responses Using Query-Aware Transformers and Neutral Reference Embeddings, in: E. S. Salido, A. B.-C. no, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026.

[94] K. L. Gwet, Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters, Advanced Analytics, LLC, 2014.