Identifying Check-Worthy Claims by Generating the Implicit Content of Enthymemes

Martial Pastor*
Centre for Language Studies
Radboud University
Nijmegen, The Netherlands
martial.pastor@ru.nl

Stefan Kok
Institute for Computing and Information Sciences
Radboud University
Nijmegen, The Netherlands
stefan.kok@ru.nl

Abstract

Fact-checking has had a tendency to overlook implicit claims, on which much misleading discourse depends. We propose a pipeline for detecting them based on the concept of the enthymeme, an argument with a missing component whose grounding in argumentation schemes allows implicit content to be recovered systematically. The pipeline proceeds in three steps: detecting whether a tweet contains enthymematic argumentation, generating the missing premise or conclusion, and assessing whether the reconstructed content constitutes a check-worthy claim. While the individual stages of the pipeline show encouraging performance, our evaluation and error analysis reveal that end-to-end claim recovery remains a challenging and open task, and we identify the main bottlenecks to guide future work.

1 Introduction

Fact-checking as an NLP task has received considerable attention in recent years (Vlachos and Riedel, 2014; Guo et al., 2022), with numerous datasets (Wang, 2017; Thorne et al., 2018; Augenstein et al., 2019) and successful systems developed (Hassan et al., 2017; Nakov et al., 2021), some now deployed to the general public (Google, 2024). Their focus, however, has been almost exclusively on *explicit* claims. This framing overlooks a frequent strategy of misleading content: false narratives are often built on unstated premises, conveying their message through what is left unsaid rather than what is asserted. Such implicit content escapes current fact-checking systems entirely, even though it could be verified if first reconstructed.

The vehicle of this strategy is the *enthymeme*: an argument with a missing component, where one of the premises or the conclusion is absent from the uttered text. Consider the tweet *"I don't wanna get a COVID vaccine because some days I would like to have babies"* (Burke and Vlah, 2023): the argument only goes through if one supplies the unstated premise that *the vaccine causes infertility* – a check-worthy claim that is never actually asserted, and that no current fact-checking system would surface. Since Aristotle, the enthymeme has been recognized as the central device of persuasion, precisely because the audience supplies the missing proposition themselves. This makes enthymemes an ideal lens for recovering the implicit claims and assumptions underlying discourse (Habernal et al., 2018; Boltužić and Šnajder, 2016) – and, conversely, a tool that can be weaponized: Ciampaglia and Licato (2021) argue that disinformation frequently presents individually accurate but selectively chosen facts while leaving the misleading belief unstated for readers to infer. These last authors call for argumentation-based fact-checking that recovers this implicit content, yet no end-to-end system exists that detects enthymematic argumentation in social media posts, generates the missing component, and assesses its check-worthiness.

We propose such a pipeline. Crucially, the missing component of an enthymeme is not arbitrary: its reconstruction is licensed by the argumentation scheme underlying the argument (Walton et al., 2008). We build on the dataset of Pastor and Oostdijk (2026), in which annotators reconstructed the implicit premises and conclusions of tweets about vaccination and immigration, grounding each reconstruction in a Walton scheme. Our pipeline comprises three stages: enthymeme detection with DeBERTa-v3 classifiers (Section 5.1), missing component generation with BART, optimized directly on the ROSCOE semantic-similarity metric (Section 5.2), and check-worthiness assessment with ClaimBuster (Section 5.3). We evaluate each stage and the full pipeline (Section 6), followed by a detailed error analysis (Section 7).

Our main contributions are:

* Corresponding author.

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 881–894
October 17-21, 2026 ©2026 Association for Computational Linguistics

- The first end-to-end pipeline surfacing implicit check-worthy claims from social media posts via argumentation scheme-grounded enthymeme reconstruction.
- A model selection methodology optimizing hyperparameters directly on ROSCOE-SS, which we show better suited to implicit content reconstruction than token-level metrics.
- An evaluation, including a comparison against a few-shot instruction-tuned LLM baseline, and a detailed error analysis delineating the key bottlenecks of the task.

2 Related Work

To the best of our knowledge, no existing work proposes an end-to-end pipeline that detects enthymematic argumentation in social media posts, generates the missing component, and assesses its check-worthiness. The need for such a system, however, has been clearly articulated: Ciampaglia and Licato (2021) argue that false narratives frequently rely on enthymemes, presenting individually accurate but selectively chosen facts while leaving the misleading belief unstated for readers to infer. Their proposed research agenda for argumentation-based fact-checking includes identifying the intended conclusion of enthymematic disinformation and using argumentation schemes to recover the premises needed to evaluate it. Our pipeline operationalizes part of this agenda: we recover both implicit premises and conclusions, our gold reconstructions are grounded in argumentation schemes, and the resulting verifiable claims can be passed to downstream fact-checking stages.

Prior work on enthymematic argumentation has addressed detection and generation separately. On the detection side, Stahl et al. (2023) predict whether a learner argument contains a missing Argumentative Discourse Unit (ADU) at a specified position, using a corpus built automatically by removing ADUs from learner essays, with DeBERTa (He et al., 2021) as a baseline. We share their decomposition of the problem into detection and generation, but our setting differs substantially: we work with short tweets in which neither the presence nor the position of the implicit component is marked, so our detector must decide whether a tweet is enthymematic at all, and our generator is conditioned on a structured input (instruction, predicted label, tweet) rather than a mask token at a known position.

On the generation side, Chakrabarty et al. (2021) recover implicit premises from enthymemes consisting of a stated premise and conclusion. Citing the scarcity of enthymeme data, they leverage the ART abductive reasoning dataset (Bhagavatula et al., 2020) as auxiliary training material and augment BART (Lewis et al., 2020) inputs with commonsense inferences from PARA-COMET (Gabriel et al., 2021). In contrast, we fine-tune BART directly on human reconstructions of enthymemes in tweets about vaccination and immigration, and our missing component may be either a premise or a conclusion, with the model inferring the appropriate type from the tweet and its predicted label alone. The two settings are complementary: theirs demonstrates the value of abductive and commonsense knowledge for premise generation, while ours covers both implicit component types in short, domain-specific social media controversial text.

As a point of reference for current performance levels, the fine-tuned DeBERTa detector of Stahl et al. (2023) reaches an F_1 of 0.68 on enthymeme detection even in their easier setting where the candidate gap position is marked in the input, of the same order as our results on tweets (Section 6.1); on the generation side, in contrast, published results are hardly comparable across works, since each is scored against its own dataset-specific references and task format.

3 Enthymemes and Claims

3.1 Enthymemes

Our methodology for extracting implicit claims draws on work at the intersection of Argumentation Theory and Argument Mining. Specifically, we rely on the definition of enthymemes formalized by Pastor and Oostdijk (2026) based on Argumentation Schemes (Walton et al., 2008).

An enthymeme is defined as an argument with a missing component: given an argument with premises p_1, p_2, and conclusion c, one of these three elements is absent from what is uttered in the natural language text (or tweet in our case). The task thus involves two steps: first identifying whether an enthymeme is present (i.e., whether an argument has a missing component), and then recovering that missing component through natural language generation. The idea of this work is to target one of these missing components which

contains fact-checkable claims.

3.2 Extracting Implicit Claims

In order to maximize the conceptual clarity of how implicit claims are recovered, we provide examples; more details on the dataset are given in Section 4.

An implicit claim is present when an enthymeme is present. Enthymemes are present only when an argument can be identified in the tweet. To justify the presence of an argument, an argumentation scheme must be identifiable. Let us consider the example below, borrowed from Burke and Vlah (2023)

> *"I don't wanna get a COVID vaccine because some days I would like to have babies"*

From the example above, we can reconstruct the following propositional breakdown, where P_2 is the reconstructed implicit claim:

P_1: I want to have babies

P_2: The vaccine causes infertility *(implicit claim)*

C: I don't want to get the COVID vaccine

The presence of an implicit premise here is justified by the fact that the *Argument from Negative Consequences* scheme can be used to recover the missing, unstated part of the argument:
The scheme *Argument from Negative Consequences* (simplified) takes the following form:

P_1: Agent X has goal G

P_2: Action A prevents goal G

C: Therefore, agent X should not perform action A

Here is another example, taken from Tweet #2481 in the dataset of Pastor and Oostdijk (2026).

> *"Ben & Jerry's... just wants cheaper labour costs and is clearly happy to see... modern slavery"*

From the example above, we can reconstruct the following propositional breakdown:

P_1: Ben & Jerry's wants cheaper labour costs

P_2: Illegal immigration/people smuggling supplies cheap labour *(implicit claim)*

C: Ben & Jerry's is happy to see people smuggled into conditions of modern slavery

The scheme *Argument from Goal* takes the following form:

P_1: Agent X wants goal G

P_2: Process/action A achieves goal G

C: Therefore, agent X supports/welcomes A

These examples illustrate how grounding enthymemes in argumentation schemes allows us to systematically recover the implicit claims underlying a tweet. Once made explicit, these claims can be submitted to fact-checking, even though they were never stated directly. In this way, we can move from a speaker's explicit utterance to the underlying assumption it relies on.

4 Data

4.1 Data Origin

To generate implicit claims for fact-checking, we make use of the implicit arguments dataset introduced by Pastor and Oostdijk (2026), which is itself built upon a preexisting dataset of tweets introduced by Flaccavento et al. (2025) in their study of trope usage in social media. In the original dataset, approximately 50,000 English language tweets were collected via keyword-based queries targeting vaccines and immigration (`"migrant"`, `"migration"`, `"asylum"`, `"vaccine"`), spanning a period of four years, from late 2019 to late 2023, without filtering for specific users. They subsequently cleaned the dataset by stripping URLs and removing personally identifiable information, without applying a profanity filter.

4.2 Dataset

From the above dataset, Pastor and Oostdijk (2026) randomly selected a subset of 1,482 tweets so as to obtain a balanced distribution of trope vs no-trope labels, which they use for enthymeme annotation, and it is this dataset and annotations that we use

for the present work. The dataset was labeled by seven annotators (five primary annotators and two validators), with every datapoint receiving at least three original annotations and two validations. Labels are imbalanced: *none* accounts for 66.1% (980 tweets), *implicit premise* for 29.6% (439), and *implicit conclusion* for 4.3% (63). Beyond the categorical label, each annotation also records a full propositional decomposition of the tweet and the argumentation scheme licensing the implicit reconstruction, as shown in Table 1 which shows what a datapoint annotation would look like per annotator.

Field	Value
Label	*Implicit premise*
P1	**Vaccines cause infertility [implicit]**
P2	I want to have babies
Conclusion	I don't want to get the COVID vaccine
Scheme	Argument from Negative Consequence

Table 1: Fields contained in a single datapoint annotation.

If we pool all annotators' reconstructions of implicit premises and conclusions, we get 1,543 reconstructed natural language sentences. However, not all of these are fact-checkable claims, some take the form of opinions, personal experience, or vague assertions that can't be verified. The number of check-worthy claims we investigate is given in section 6.3.

5 Implicit Factual Claim Detection Pipeline

In this section, we outline the steps of our enthymeme-based pipeline for implicit claim detection in tweets[1], summarized in Figure 1.

5.1 Enthymeme Detection

We model enthymeme detection as a binary text-classification task. Given a tweet, the model predicts whether the text contains enthymematic argumentation or not. Related work by Stahl et al. (2023) w.r.t enthymeme detection (as mentioned in Section 2) showed that the DeBERTa-base model (He et al., 2021) achieved strong results when compared to a random and majority baseline model. DeBERTa is a suitable backbone for this task as its disentangled attention mechanism models word content and relative position separately, which can help the classifier assess whether the explicitly stated parts of a short argument form a coherent premise-conclusion relation or leave a reasoning gap. As such, we adopt DeBERTa-v3 (He et al., 2023), which improves upon the original DeBERTa architecture (He et al., 2021), as our backbone and fine-tune both the base and large variants. As our classification task is affected by a class imbalance, model selection and evaluation are based on macro F1-score, which gives equal weight to both classes. We keep a held out test set untouched during model development whereas the development set is split into a training portion and a dev-test portion using an 80/20 split.

Hyperparameter optimization (HPO) constitutes the methodological choice we experiment with in this work, and is performed exclusively on the training portion. The dev-test portion is reserved for final model selection procedures, including validation during a final training run and threshold tuning. Final performance is reported only on the untouched test set.

5.1.1 Model Construction Pipeline

Figure 3 (in Appendix A) summarizes the model construction pipeline used for both DeBERTa-v3-base and DeBERTa-v3-large. In the first step we use Optuna (Akiba et al., 2019) to optimize the hyperparameters of the classification models. The optimized hyperparameters are the learning rate, weight decay, number of warmup steps, training batch size and the number of epochs. The Optuna objective is to maximize the mean-macro-F1 score which is obtained through a stratified 5-fold cross validation on the training split.

In each Optuna trial (see Appendix A.1 for a high level overview of each Optuna trial), the Tree-structured Parzen Estimator sampler proposes a candidate hyperparameter configuration. A fresh DeBERTa model is then initialized and fine-tuned with this configuration for each fold. The macro-F1 scores across the 5 validation folds are averaged and returned to Optuna as the trial score. This cross-validation setup reduces the dependence on a single validation split and makes the selected hyperparameters less sensitive to a particular partition of the data. In the second step, the best hyperparameter configuration found by Optuna is used to train final models from fresh initializations. We train one single model, and separately, three models initialized with different random seeds. The latter is used to construct an ensemble by averaging the predicted class probabilities.

[1]The full code and data samples can be accessed at https://github.com/metabolean5/inlg-enthymemes.

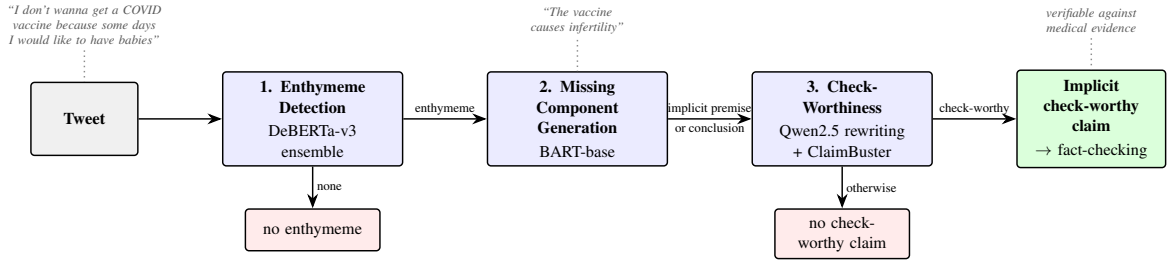


Figure 1: The three-stage pipeline for surfacing implicit check-worthy claims, illustrated with the running example from Section 1. Red boxes mark the exit points at which a tweet is predicted to carry no implicit check-worthy claim.

In the third step, we tune the decision threshold on the dev-test split, both for the single model and for the ensemble. The resulting threshold-tuned single model and the threshold-tuned ensemble are then evaluated on the untouched test set. This pipeline allows us to compare whether the additional robustness of ensembling improves performance over a single fine-tuned DeBERTa model.

5.2 Missing Component Generation

We model the generation of the implicit component as a conditional text-generation task. The model receives a structured input containing the task instruction, the predicted label (indicating whether the tweet contains enthymematic argumentation, see Appendix B.2), and the tweet text, and generates either the implicit argument or a sentinel token. The input format is:

`task: generate implicit_argument | label: <argument | none> | tweet: <tweet_text>`

We fine-tune BART-base (Lewis et al., 2020) for this task. We also experimented with BART-large but found it unstable in our setting: with higher learning rates, training loss sometimes spiked before collapsing to zero, and after lowering the learning-rate range, the loss remained high without meaningful decrease, with outputs that were often empty, sentinel-only, or malformed. We therefore retained only BART-base for the final generation experiments.

For tweets without enthymematic argumentation, we define a sentinel token `<NO_IMPLICIT>`, allowing the model to explicitly learn when no implicit argument should be generated. We further expand the dataset introduced in Section 4.2 into long format: for each tweet whose majority label is premise or conclusion, we create one data point per available reconstruction (see Appendix B.1).

For hyperparameter optimization we reuse the Optuna-based strategy of the classifiers, tuning the learning rate, weight decay, warmup steps, training batch size, and number of epochs, with five-fold cross-validation inside each trial.

A key aspect of our method lies in the optimization objective: Optuna maximizes the ROSCOE semantic-similarity score (Golovneva et al., 2023) between the generated text and the available gold reconstructions (see Appendix C.1). As we show in Section 6.2, ROSCOE-SS is the metric best suited to our task, so optimizing hyperparameters directly on it aligns model selection with the property we ultimately care about. Examples whose references contain only the sentinel token are excluded from the score, and decoding hyperparameters are kept fixed at their default values during HPO and the final run.

Unlike the binary classification setting, BART produces a text sequence rather than a class probability, so there is no decision threshold to tune. We also do not construct an ensemble, since ensembling autoregressive sequence-to-sequence models would substantially increase training and decoding costs. Instead, we fine-tune a single BART-base model with the best hyperparameter configuration and evaluate it on the dev-test set.

5.3 Fact-Checkable Claim Detection

The final stage of the pipeline determines whether the generated implicit content constitutes a verifiable claim. We pass each generated reconstruction to ClaimBuster (Hassan et al., 2017), which scores its check-worthiness. Reconstructions classified as check-worthy are predicted to contain a fact-checkable claim; all other outputs, including the sentinel token, are predicted as containing no claim. Since annotators often phrased reconstructions as conditionals, a style BART learned to replicate and which rarely reads as a checkable claim,

we first rewrite the generated outputs into direct declarative propositions using few-shot Qwen2.5-7B-Instruct, preserving the original verbs and entities (see Section 6.3 for details). The claims surfaced by this stage can then be passed to downstream fact-checking systems.

Note that Qwen2.5-7B-Instruct (Qwen Team et al., 2025) plays two distinct roles in this work: here as a surface-form rewriter applied to both gold and system outputs, and in Section 6.3 as a few-shot detect-and-generate baseline against which we compare the pipeline.

6 Evaluation and Results

We evaluate each stage of the pipeline independently against held-out gold labels: binary detection (enthymeme vs. no enthymeme), implicit-content generation, and downstream claim verification. The held-out test set contains 148 tweets (97 without and 51 with implicit content).

6.1 Detection

Table 2 reports held-out performance of the two 3-seed ensembles detailed in Section 5.1.1; the large ensemble (threshold 0.53) reaches the best macro-F_1 of 0.686 (vs. 0.662 for the base ensemble) and feeds the downstream stages. This is a solid margin over the no-skill floor for this $\sim$2:1 imbalanced binary task, though the gap to the *none* class F_1 (0.76 vs. 0.61) shows implicit content remains the harder class. Both ensembles over-predict the minority class: recall on *implicit* (0.69) consistently exceeds precision (0.52–0.56).

6.2 Implicit Content Generation

To evaluate the implicit content generations, we score outputs with ROSCOE-SS (Golovneva et al., 2023), a semantic similarity metric from a suite originally designed to assess the quality of step-by-step reasoning. Unlike token-level metrics, it compares hypothesis and reference as whole texts through a reasoning-tuned sentence encoder, yielding a score in $[0, 1]$; here we compare each generated implicit component against the gold annotator reconstructions of the same tweet.

We evaluate our BART-base model introduced above against the 211 gold reconstructions of the test set (3–5 per tweet, spread over 50 tweets), drawn from the annotated set where annotators made the implicit premise or conclusion explicit. Since generation only runs on tweets the detector flags as enthymematic, scores are computed over the 26 test tweets for which both a generated hypothesis and gold reconstructions are available. Since raw scores occupy a narrow band and are hard to interpret, we anchor them between a random floor (hypotheses re-scored against 20 wrong-tweet reference permutations) and a human ceiling (pairwise inter-annotator agreement, 3–5 annotators $\times$ 50 tweets); this interval is the metric's effective range for our task (Table 3). The model reaches 0.900, covering 62% of the distance from the random floor (0.847) to the human ceiling (0.929). A BERTScore (Zhang et al., 2020) cross-check preserves the ordering but nearly halves the model–random gap (+0.029 vs. +0.053): its token-level matching rewards shared in-domain vocabulary even when the underlying claim is wrong, inflating the floor. ROSCOE-SS thus discriminates genuine reconstruction from topical noise more sharply, making it the better-suited metric here.

6.3 Downstream Claim Verification and Full Pipeline Evaluation

Finally, we test whether the generated implicit content constitutes a *verifiable claim*, using ClaimBuster (Hassan et al., 2017). Gold labels come from the annotator reconstructions of the test set: 211 reconstructions (3–5 per tweet) spread over 50 tweets. Running ClaimBuster on these reconstructions and manually verifying its judgements confirms 39 of them as check-worthy; since a single claim-bearing reconstruction suffices, this yields 23 tweets containing gold implicit check-worthy claim on the 148-tweet test set (23 claim / 125 no-claim).

We then apply the same procedure to the system side[2]: each tweet is run through the full pipeline, i.e. the detection ensemble, followed by BART generation for tweets flagged as enthymematic, followed by ClaimBuster on the generated implicit content. As an instruction-tuned LLM point of comparison, we also evaluate a *Qwen detect+generate* variant, in which few-shot prompted Qwen2.5-7B-Instruct replaces the first two stages: given the tweet and four in-context examples, it jointly decides whether an implicit argument component is present and, if so, generates it as a declarative proposition; the ClaimBuster stage is unchanged. A check-worthy

[2]Some annotators habitually phrased reconstructions as conditionals, a style BART learned to replicate; since conditionals rarely read as checkable claims, we first rewrite both gold reconstructions and system outputs into direct declarative propositions with few-shot Qwen2.5-7B-Instruct, preserving the original verbs and entities.

	None			Implicit				
Ensemble (3 seeds)	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**Acc.**	**Macro-F_1**
DeBERTa-v3$_{base}$	0.80	0.67	0.73	0.52	0.69	0.59	0.676	0.662
DeBERTa-v3$_{large}$	**0.81**	**0.71**	**0.76**	**0.56**	0.69	**0.61**	**0.703**	**0.686**

Table 2: Binary detection on the held-out test set (148 tweets). The large ensemble is used in the deployed pipeline and feeds the generation and claim-verification stages.

System	**ROSCOE-SS**	**BERTScore**
Human ceiling (IAA)	0.929	0.920
BART pipeline	0.900	0.893
Random floor ($\pm$0.004)	0.847	0.864

Table 3: Generation quality on 26 matched test tweets, with human ceiling and random-permutation floor.

System	**Cov.**	**P**	**R**	**F_1**	**Acc.**
Full pipeline	67	**0.32**	0.35	**0.33**	**0.78**
Qwen detect+generate	138	0.15	**0.39**	0.22	0.57

Table 4: End-to-end claim verification on the 148-tweet test set (23 gold claim / 125 no-claim). Full pipeline = detection ensemble + BART generation + ClaimBuster. P/R/F_1 are reported on the claim class; Cov. = number of tweets for which the system generated implicit content (remaining tweets default to no-claim).

classification predicts *claim*, anything else, including the absence of generated content, predicts *no-claim*. Matching these predictions against the gold labels gives the claim-class precision, recall, and F_1 in Table 4. Coverage, the share of tweets that reach the generation stage, is the bottleneck: the pipeline produces content for 67 of 148 tweets, recovering 8 of 23 gold claims, while the Qwen detect-and-generate variant over-generates (138/148 tweets), trading precision (0.15) for the highest recall. This full-set ranking partly reflects the pipeline's narrower coverage aligning with the majority no-claim class: on the 51 annotator-covered tweets (23 claim / 28 no-claim), where defaulting to no-claim yields no easy wins, the Qwen variant reaches F_1 0.44.

6.4 Overall Results

Detection is the most robust stage, with macro-F_1 of 0.66–0.70 across cross-validation and held-out settings, well above the no-skill floor for this $\sim$2:1 imbalanced task. Generation, as measured by ROSCOE-SS, produces output that clearly rises above topical noise, covering roughly two thirds of the distance from the random floor to the human ceiling, a solid but not yet human-level performance. Claim verification, finally, shows how upstream errors compound: with claim-class F_1 capped at 0.33 by detection coverage, the end-to-end task remains largely open. To understand these losses better, the next section analyses what is lost, and why, at each step of the pipeline.

7 Error Analysis

To understand where the pipeline loses claims, we manually inspect all errors on the test set: of the 23 gold claims, the pipeline recovers 8, misses 15, and additionally produces 17 false positives. The misses split evenly between two stages: 7 tweets are discarded by the detector before anything is generated, and 8 are generated but rejected by the claim classifier. We examine each failure mode in turn, then the false positives, and close with the successful cases.

7.1 Detection Failures

Seven gold claims are lost at the first step: the detector labels the tweet as non-enthymematic, so no implicit content is ever generated. A manual analysis of these seven tweets, which concerns only the cases where the implicit content is a check-worthy claim and makes no claim about enthymeme detection in general, shows they lack any explicit inferential connector (*because*, *but*, *this will result in*): their argumentative move is carried through juxtaposition, enumeration, or register, giving the detector no surface structure to latch onto. Tweet #225 illustrates this:

> *"I would be very skeptical of these results. It's easy to skew data collection [...] and the pharma companies and regulators have every motivation to make the vaccine look better than it actually is."*

Pipeline: `<NO_IMPLICIT>`
Gold: Pharma manipulates vaccine data

Because these losses occur before generation, they are unrecoverable downstream: detection recall directly caps end-to-end recall.

7.2 Generating the Normative Instead of the Factual

The remaining eight misses are subtler: the detector fires, BART generates implicit content, but ClaimBuster labels it a non-factual statement. Inspection shows the classifier is usually right: the generated reconstructions are moral or normative ("it is wrong to…", "you are being manipulated"), while the annotators identified a *factual* implicit claim in the same tweet. Consider Tweet #1522:

> *"Still clinging to the saving the world by getting injected with an experimental 'vaccine' narrative huh? [...] How does it feel being manipulated?"*

Pipeline:	You are manipulated into doing something not in your interest
Gold:	The government lied about the vaccine's effectiveness

The pipeline stops at the normative surface of the argument, while annotators identify the factual proposition supporting it. These are therefore generation failures masquerading as classification failures: the claim classifier correctly rejects what it receives, but what it receives is the wrong reconstruction. This matters for our downstream goal, since it is precisely the factual layer that fact-checking requires.

7.3 Overgeneration

The 17 false positives fall into different modes of increasing severity. In the mildest, the pipeline lifts an assertion that is already *explicit* in the tweet (e.g. Tweet #2655: "free movement depressed wages in several sectors", copied nearly verbatim) and presents it as implicit; the output is genuinely check-worthy, it just was never hidden.

In the most severe mode, the pipeline hallucinates from topic salience, as in Tweet #837:

> *"Sen. Graham Says Economy Will Trump Abortion in November Elections | so true!! [...] Vaccine mandate Rem?"*

Pipeline:	Vaccine mandates harm people's health
Gold:	no check-worthy claim

Here the tweet *is* enthymematic, and the detector rightly fires: annotators reconstructed an implicit normative argument about bodily autonomy, none of which was classified as check-worthy. The failure lies in generation. Instead of producing this non-claim reconstruction, BART, potentially triggered by the phrase "vaccine mandate", generates a common factual claim from its training distribution that appears nowhere in the tweet, implicitly or otherwise. The generated sentence is coherent and check-worthy in isolation, so ClaimBuster correctly labels it a claim. The real problem at this stage is thus the confusion between generating claims and generating non-claims: when the gold implicit content is not a claim, BART should generate something closer to the annotators' non-claim reconstructions, and the claim classifier has no signal to catch it when it does not.

7.4 Successful Cases

Finally, the eight true positives show the pipeline working as intended, but with caveats. In the best case, Tweet #1205, the pipeline converges with all five annotators:

> *"So cool how they now have another 'flu' that will require a distinct vaccine EVERY year! What a windfall for the industry!"*

Pipeline:	Vaccine mandates are a windfall for the industry
Gold:	The vaccine industry profits from recurring vaccines

Even among true positives, however, the generated text often differs from the gold reconstructions in ways the claim-level match hides. In Tweet #376, the pipeline extracts an embedded factual sub-claim ("vaccines are made from fetal cells") where annotators identified the hypocrisy argument built on it; in Tweet #2160, the generated implicit is logically *inverted* relative to what annotators reconstructed, yet still passes as a check-worthy claim. A binary claim/no-claim match therefore overstates fidelity: the pipeline can land on *a* check-worthy claim in the right tweet without recovering *the* claim the argument actually relies on.

8 Discussion and Conclusion

We presented a first attempt at detecting implicit check-worthy claims in controversial tweets, through a pipeline combining enthymeme detection, missing component generation, and check-worthiness assessment. Grounding reconstructions in argumentation schemes provides a systematic justification for the recovered implicit content, and optimizing the generation model directly on ROSCOE-SS aligns model selection with the property the task requires.

While the individual stages perform well above chance, end-to-end claim recovery remains capped at F_1 0.33, with losses compounding across stages. The detector struggles with tweets whose argumentative move is carried without inferential connectors; when it fires, the generator often reconstructs the normative surface of the argument where annotators dig one level deeper to the factual proposition supporting it, precisely the layer fact-checking requires. Overgeneration, conversely, is driven by topic salience, with plausible in-domain claims occasionally generated that appear nowhere in the tweet.

At the same time, our results demonstrate that the approach is viable: the pipeline does succeed in surfacing genuinely implicit check-worthy claims that would escape standard fact-checking entirely, and its detection and generation stages already show promising performance. Notably, the fine-tuned pipeline outperforms the few-shot instruction-tuned LLM baseline end-to-end, which attains higher recall but at a substantial cost in precision (Table 4). These findings delineate clear directions for future work: modeling argumentation cues in detection, distinguishing factual from normative implicit content in generation, and moving evaluation beyond binary claim matching, which can overstate fidelity when the pipeline lands on *a* check-worthy claim without recovering *the* claim the argument relies on. This first pipeline, its promising results, together with its documented failure modes, provide a foundation for subsequent work on argumentation-based fact-checking.

Acknowledgments

This work was produced as part of the HYBRIDS project, a Marie Skłodowska-Curie Doctoral Network funded by the European Union under grant no. 101073351 and UKRI Horizon Funding Guarantee, and the AI-CODE project under Horizon Europe grant agreement no. 101135437.

Limitations

Our study faces several limitations. First, the dataset is small: 1,482 annotated tweets, of which only 148 form the held-out test set, with 23 gold check-worthy claims. Results at the claim level therefore rest on few positive instances, and small absolute differences should be interpreted with caution. For the same reason, we report point estimates without confidence intervals or significance tests: with so few positive instances, such intervals would be too wide to be informative, and our scores should be read as indicative rather than definitive. The class imbalance, particularly the scarcity of implicit conclusions (4.3% of the data), further limits what the models can learn about this component type.

Second, the multi-step architecture makes the pipeline vulnerable to error propagation. Tweets discarded by the detector never reach the generation stage, so detection recall directly caps end-to-end recall, and generation errors are inherited by the claim classifier, which has no access to the original tweet to catch unfaithful reconstructions. This cascading structure means that stage-level improvements do not translate linearly into end-to-end gains.

Third, the data covers two topics (vaccination and immigration) in English tweets from a specific time period, and it is unclear how well the pipeline generalizes to other domains, languages, platforms, or argumentative styles. The reliance on ClaimBuster as check-worthiness classifier and on ROSCOE-SS as optimization and evaluation metric also ties our results to the behavior of these specific tools, and we do not ablate their individual contributions: in particular, the Qwen-based declarative rewriting step is applied to both gold and system outputs, so its effect on end-to-end scores is not separately quantified. Likewise, our LLM comparison is limited to a 7B-parameter instruction-tuned model; a demonstration of the potential of larger, state-of-the-art LLMs on this task remains future work.

Finally, our pipeline rests on the assumption that a tweet's implicit content can be recovered as a single ground truth. Yet enthymeme detection, let alone reconstruction, involves a degree of semantic interpretation, and what a controversial post might imply can vary across individuals and communities; a system that announces what speakers meant but did not say would not go without reminding us of the concept of a police of thought that a certain George Orwell once wrote about – a concern echoed by recent critiques of detection-centered misinformation research, which treats such label variation as noise to suppress rather than signal to model (Bassi et al., 2026). Like most work in this area, we follow the standard practice of training on majority labels and pooled reconstructions; a natural next step for pipelines of this kind is to also incorporate label-variation-oriented metrics

that measure disagreement rather than averaging over it (Plank, 2022; Jiang and de Marneffe, 2022). We take a first step in this direction by studying annotator disagreement in enthymeme annotation in Pastor and van Arkel (2026).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, pages 2623–2631, Anchorage, AK, USA. Association for Computing Machinery.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Davide Bassi, Søren Kirkegaard Fomsgaard, Erik Bran Marino, and Katarina Laken. 2026. Combating disinformation: Is there no alternative? In *Proceedings of the 1st Workshop on Information Disorder (InDor) @ LREC 2026*, pages 1–15, Palma de Mallorca, Spain. ELRA Language Resources Association (ELRA).

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *Proceedings of the International Conference on Learning Representations*.

Filip Boltužić and Jan Šnajder. 2016. Fill the gap! Analyzing implicit premises between claims from online debates. In *Proceedings of the Third Workshop on Argument Mining (ArgMining 2016)*, pages 124–133, Berlin, Germany. Association for Computational Linguistics.

Michael Burke and Ana Vlah. 2023. Teaching the enthymeme: On becoming a critical-rhetorical consumer of verbal and visual arguments. In D. Tomić, J. Vlašić Duić, and E. Pletikos Olof, editors, *Rhetorical Research and Didactics: RHEFINE. Rhetoric for innovative education*, pages 37–55. University of Warsaw, Warsaw.

Tuhin Chakrabarty, Aadit Trivedi, and Smaranda Muresan. 2021. Implicit premise generation with discourse-aware commonsense knowledge models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6247–6252, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Giovanni Luca Ciampaglia and John Licato. 2021. Fact-checking, false narratives, and argumentation schemes. In *Proceedings of the 1st International Workshop on Knowledge Graphs for Online Discourse Analysis (KnOD 2021) co-located with the 30th The Web Conference (WWW 2021), Virtual Event (due to COVID-19 outbreak), April 14, 2021*, volume 2877 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Alessandra Flaccavento, Youri Peskine, Paolo Papotti, Riccardo Torlone, and Raphael Troncy. 2025. Automated detection of tropes in short texts. In *Proceedings of the 31st International Conference on Computational Linguistics*, Abu Dhabi, UAE. Association for Computational Linguistics.

Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. 2021. Paragraph-level commonsense transformers with recurrent memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12857–12865.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *Proceedings of the International Conference on Learning Representations*.

Google. 2024. Fact check tools. `https://toolbox.google.com/factcheck/explorer`. Accessed July 2026.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by ClaimBuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *Proceedings of*

the International Conference on Learning Representations.

Nan-Jiang Jiang and Marie-Catherine de Marneffe. 2022. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Watheq Mansour, Bayan Hamdan, Zien Sheikh Ali, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, Thomas Mandl, Mucahid Kutlu, and Yavuz Selim Kartal. 2021. Overview of the CLEF–2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: Proceedings of the 12th International Conference of the CLEF Association*, pages 264–291. Springer.

Martial Pastor and Nelleke Oostdijk. 2026. A resource for enthymeme detection in controversial political discourse. *arXiv preprint arXiv:2606.12186*.

Martial Pastor and Jarno van Arkel. 2026. Disagreement in enthymeme annotation. In *Proceedings of the MediaEval 2026 Multimedia Evaluation Workshop*, CEUR Workshop Proceedings, Amsterdam, Netherlands and Online. CEUR-WS.org. To appear.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, and 1 others. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Maja Stahl, Nick Düsterhus, Mei-Hua Chen, and Henning Wachsmuth. 2023. Mind the gap: Automated corpus creation for enthymeme detection and reconstruction in learner arguments. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4703–4717. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

Douglas Walton, Chris Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press, Cambridge, UK.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *The Eighth International Conference on Learning Representations (ICLR)*.

A Model Construction Pipeline

A.1 Optuna Trial Overview

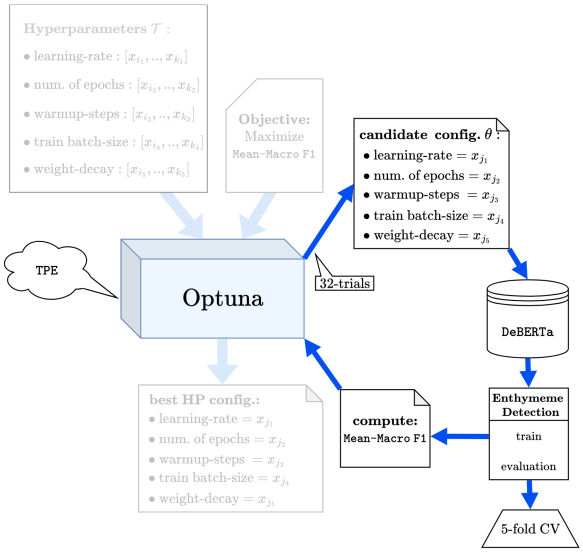


Figure 2: High-level overview of an Optuna trial for construction of classification models.

Figure 2 illustrates the procedure for the enthymeme detection models, but the same optimization procedure is used for the implicit argument

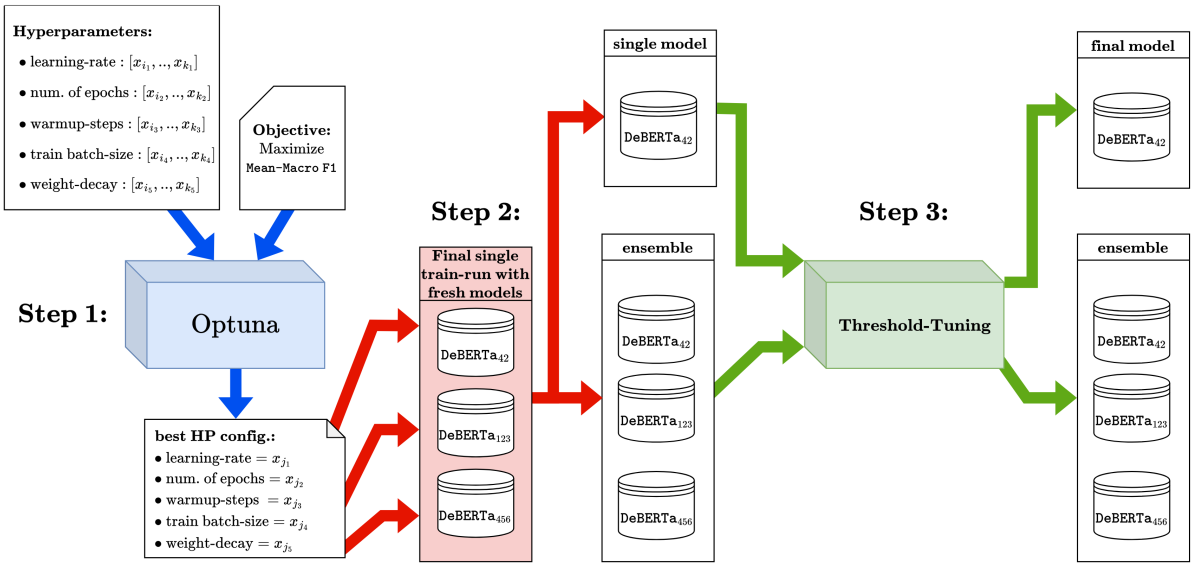


Figure 3: Pipeline for building the models for enthymeme detection.

generation model. Let

$$\mathcal{T} = \{\text{learning rate, weight decay, warmup steps, number of epochs, batch size}\}.$$

be the set of hyperparameters optimized by Optuna. Then each hyperparameter is associated with a predefined search space. For the classification models, the objective is maximizing the mean macro F1-score obtained through stratified five-fold cross validation on the training split. We used Optuna's default Tree Structured Parzen Estimator (TPE) sampler. At the start of each trial TPE proposes a candidate hyperparameter configuration

$$\theta = \{x_{j_1}, x_{j_2}, .., x_{j_5}\},$$

where $x_{j_1}, ..x_{j_5}$ denote the sampled hyperparameter values for each of the corresponding hyperparameters in $\mathcal{T}$.

The sampled configuration is then evaluated by instantiating a fresh DeBERTa model and fine-tuning it under the proposed hyperparameter values. Since we use five-fold cross-validation this procedure is repeated across five folds where in each fold, four parts of the training split are used for fine-tuning and the remaining part is used for validation. The macro-F1 scores are obtained from five validation folds and then averaged yielding the mean macro F1-score for the trial. This value is then returned to Optuna as the trial objective. Optuna uses the accumulated history of previous trials and their objective values to guide the TPE sampler towards more promising regions of the search space in subsequent trials.

As it is hard to find a standard procedure for determining the optimal number of trials, we set the total number of trials to $2^{|\mathcal{T}|}$ since this is a practical compromise between search space coverage and computational costs.

For the BART-base missing component generation model the same HPO trial structure is used, but the objective differs. Instead of training a DeBERTa classifier and maximizing the mean macro F1-score, each trial trains a fresh BART-base model using the sampled hyperparameter configuration and evaluates the outputs with the ROSCOE-SS. The five-fold ROSCOE-SS scores are averaged and returned to Optuna as the objective value. The decoding hyperparameters are kept fixed during HPO so Optuna only optimizes the hyperparameters in $\mathcal{T}$. Thus the classification and generation HPO procedures differ in the model architecture and objective metric but share the same trial structure, set of hyperparameters, five-fold cross-validation scheme and TPE-based sampling procedure.

B Missing Component Generation Dataset

B.1 Long Format Specification

To fine-tune BART-base for missing component generation we transform the original dataset introduced in Section 4.2 into a long-format generation dataset. In the original dataset, each tweet is associated with five annotator specific label fields

ann$_i$_label, and five corresponding reconstruction fields, ann$_i$_implicit, with $i \in \{1, \ldots, 5\}$. Each annotator labels the tweet as containing an implicit premise, conclusion or no implicit component, and provides a reconstructed implicit text when applicable.

For each tweet we use the majority label as the target label. If the majority label is either premise or conclusion, we create one long-format row for every annotator whose label agrees with the majority label. Each such row preserves the original tweet id and tweet_text but replaces the annotator specific fields with a single target field, implicit_text, containing the annotator's reconstruction. As a result, each single tweet may yield multiple training rows, one for each majority agreeing reconstruction.

If the majority label is none, we create a single row for the tweet and set implicit_text = <NO_IMPLICIT> where NO_IMPLICIT is the sentinel token indicating that no implicit component should be generated.

After constructing the long-format dataset we split it into a training (used for HPO) and dev-test split. When splitting we ensure all long format rows with the same tweet id are assigned to the same split such that no different reconstruction of the same tweet appears in both splits. The training portion consists of 2136 datapoints corresponding to 1066 unique tweet ids. This format allows the model to learn multiple valid reconstructions of the same enthymematic tweet while preserving a single prediction target per training row.

B.2 Structured Input

In Section 4.2 and Appendix B.1, we described the original and long-format datasets as using a three-class majority label: premise, conclusion, or none. However, the structured input for BART in Section 5.2 contains a binary label. This is because the generation model comes after the enthymeme detection model in the pipeline, whose output is a binary label indicating whether the tweet contains enthymematic argumentation or not. We therefore use a mapper f mapping the three-class majority label ℓ to a binary label:

$$f(\ell) = \begin{cases} \texttt{argument}, & \text{if } \ell \in \{\texttt{premise}, \\ & \qquad \texttt{conclusion}\}, \\ \texttt{none}, & \text{if } \ell = \texttt{none}. \end{cases}$$

The resulting binary label is then used in the structured input for BART (Section 5.2). At inference time, detector outputs labeled enthymeme are converted to argument before constructing the BART input, while none is kept unchanged. This only aligns the label names used by the detection and generation stages. As a result, BART is not given the fine-grained missing-component type, but instead is only told whether the tweet contains an implicit component or not. This makes the generation setup consistent with the preceding binary enthymeme detection stage.

C ROSCOE Semantic Similarity

C.1 Computing the ROSCOE-SS Score

To compute the ROSCOE-SS Semantic Coverage Chain score on a validation fold, we first transform the long-format validation fold into a grouped validation fold. That is, all rows with the same tweet id, and therefore the same tweet_text, are collapsed into a single validation example. The corresponding implicit_text values are collected into a reference set

$$R_d = \{r_{d,1}, \ldots, r_{d,j}\},$$

where $r_{d,i}$ denotes the i-th gold reconstruction for grouped validation example d, and $j = |R_d|$ is the number of available reconstructions for that tweet.

For each grouped validation example d, BART generates a single hypothesis h_d. We compute the semantic similarity between h_d and each reference $r_{d,i} \in R_d$ using the ROSCOE encoder. Let $\mathbf{e}(x)$ denote the normalized embedding of text x. The reference-level score is computed as the cosine similarity between the generated text and the gold reconstruction, rescaled to the interval $[0, 1]$:

$$s(r_i, h_d) = \frac{1 + \cos(\mathbf{e}(r_i), \mathbf{e}(h_d))}{2}.$$

The ROSCOE-SS score for a grouped validation example is then the mean over all available non-sentinel references:

$$\text{ROSCOE-SS}(d) = \frac{1}{|R_d|} \sum_{r_i \in R_d} s(r_i, h_d).$$

Examples whose reference set contains only the sentinel token <NO_IMPLICIT> are excluded from the ROSCOE-SS computation, since there is no implicit text whose semantic content can be compared. If the model generates the sentinel token for

an example that has real implicit-text references, the example receives a score of 0. Sentinel behavior is evaluated separately by computing sentinel accuracy on examples whose majority label is none.

The ROSCOE-SS score for the validation fold V is then the mean over all scorable grouped validation examples:

$$\text{ROSCOE-SS}(V) = \frac{1}{|V'|} \sum_{d \in V'} \text{ROSCOE-SS}(d),$$

where V' is the subset of grouped validation examples whose reference set is not sentinel-only. In five-fold cross-validation, this score is computed once per fold. The Optuna objective for BART during HPO is the mean of the five-fold ROSCOE-SS scores.