big data and cognitive computing

MDPI

Article

Cognitive Computing Frameworks for Scalable Deception Detection in Textual Data

Faiza Belbachir

LYRIDS ECE-Paris, 10 Rue Sextius Michel, 75015 Paris, France; fbelbachir@ece.fr

Abstract

Detecting deception in emotionally grounded natural language remains a significant challenge due to the subtlety and context dependence of deceptive intent. In this work, we use a structured behavioral dataset in which participants produce truthful and deceptive statements under emotional and social constraints. To maintain label accuracy and semantic consistency, we propose a multilayer validation pipeline combining self-consistency prompting with feedback-guided revision, implemented through the CoTAM (Chain-of-Thought Assisted Modification) method. Our results demonstrate that this framework enhances deception detection by leveraging a sentence decomposition strategy that highlights subtle emotional and strategic cues, improving interpretability for both models and human annotators.

Keywords: deception detection; cognitive computing; computational framework; reasoning pipeline; algorithmic scalability; information systems; veracity classification

1. Introduction

Lying and misleading others are common in human communication, especially in emotionally charged or socially complex situations. People often rely on intuition, context, and subtle emotional signals to recognize when someone is not being truthful. For machines, detecting these untruths is much harder. In natural language processing (NLP), truthful and misleading messages can appear very similar in wording and sentence structure. Existing datasets are often small, artificial, or lack emotional and contextual depth, which makes it difficult for current models to perform well in real-world scenarios.

To address these issues, we use the dataset introduced by von Schenk et al. [1], where participants produced both truthful and deceptive messages in interactive and realistic conditions. Each message is labeled not only for whether it is true or false but also for how truthful it appears to others. This allows us to study deception in more detail than with simple true/false labeling. From a computer science perspective, our work introduces a new computational framework for deception detection. Instead of treating the problem only as an NLP challenge, we design an architecture that combines semantic analysis, reasoning, and scalability. This makes our approach suitable for large-scale systems where deception detection is part of broader information processing pipelines.

Our contributions are as follows:

- Semantic Decomposition: We break down each message into smaller meaning units, making it easier to capture emotional and structural patterns linked to deception.
- Reflective Validation with CoTAM: We design a step-by-step reasoning pipeline where large language models (LLMs) check and refine their own judgments, improving accuracy in deception detection.

check for updates

Academic Editor: Giuseppe Maria Luigi Sarnè

Received: 28 August 2025
Revised: 2 October 2025
Accepted: 11 October 2025
Published: 14 October 2025

Citation: Belbachir, F. Cognitive Computing Frameworks for Scalable Deception Detection in Textual Data. *Big Data Cogn. Comput.* **2025**, *9*, 260. https://doi.org/10.3390/bdcc9100260

- Modular Framework: We present a flexible architecture where each reasoning step can be scaled or optimized independently, aligning with principles of cognitive computing.
- Compositional Analysis Strategy: Our approach identifies and organizes emotional and semantic cues while remaining efficient enough for distributed computing environments.
- Evaluation: We combine standard quantitative metrics represented by F1 and Macro-F1 with qualitative analysis of emotional and semantic coherence in model outputs.

When focusing on scalability, flexibility, and systematic design, our framework not only improves deception detection performance but also supports real-world applications such as misinformation control, fraud detection, and digital trust systems.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 describes the dataset. Section 4 presents the self-verification methodology. Section 5 outlines validation strategies. Section 6 reports experimental results, followed by discussion and conclusion.

2. Related Work

Deception detection has been approached from multiple angles, ranging from early psycholinguistic analyses to recent transformer-based and multimodal systems. These approaches highlight the interdisciplinary nature of the problem but also reveal that there is no consensus on which representations are most reliable across contexts. In particular, while NLP-driven methods have advanced rapidly, many studies remain limited by dataset design choices that reduce ecological validity. To situate our study within this evolving landscape, we organize the literature review into four key areas: (1) deception detection in text-based NLP, (2) deception in contextual and dialogic settings, (3) corpora and cross-domain generalization, and (4) multimodal and physiological approaches. This structure highlights how linguistic features have been explored in both constrained and naturalistic contexts, and how methodological choices from dataset design to modality selection affect performance and generalizability. It underscores the gap that motivates our contribution: the need for methods that are both linguistically grounded and computationally scalable.

2.1. Deception Detection in Text-Based NLP

Early computational work revealed systematic differences between truthful and deceptive texts. Newman et al. [2] showed that deceptive writing tends to feature fewer self-references and more negative emotion words, reflecting cognitive distancing. Vrij [3] emphasized the increased cognitive load and syntactic complexity associated with lying. In online contexts, Hancock et al. [4] found that liars in dating profiles suppress negative emotions and emphasize positivity to appear more trustworthy. Levitan et al. [5] extended these findings to interview transcripts using linguistic and psycholinguistic features (LIWC, pronouns, sentiment), achieving 75Transformer models now dominate the field. Barsever et al. [6] proposed a BERT + BiLSTM + attention framework, achieving 93.6% accuracy on the Ott corpus. Ilias et al. [7] evaluated fine-tuned RoBERTa and DistilBERT models with LIME, reaffirming the predictive power of affective and cognitive markers. However, most transformer-based approaches are evaluated on narrow corpora and often fail to generalize to unseen settings, which raises questions about their robustness beyond benchmark conditions.

2.2. Deception in Contextual and Dialogic Settings

Deceptive behavior is highly sensitive to context: social stakes, audience, and prior discourse shape how lies are composed and interpreted. In job interviews, Levitan et al. [5] observed that deceptive candidates use fewer self-references and adopt a more formal tone

to appear sincere. Toma and Hancock [8] found similar distancing strategies in dating profiles. Fornaciari and Poesio [9] incorporated discourse-level features such as coherence and argument structure into deception detection in court transcripts, improving performance over lexical-only baselines. Game-based studies highlight strategic deception in interactive dialogue. Niculae and Danescu-Niculescu-Mizil [10] analyzed betrayal in the game Diplomacy, revealing the linguistic precursors to deceptive acts. Peskov et al. [11] expanded on this with a dual-labeled corpus (truth + perceived truth), allowing the modeling of both the intent of the speaker and the interpretation of the listener. More recently, Baravkar et al. [12] proposed a BERT-based decision engine that integrates proximity of linguistic markers (e.g., hedges, negations), achieving an improvement 5% over traditional models. These studies confirm that deception cannot be studied in isolation from its context, but they also show that dialogic realism is often sacrificed for annotation convenience. In contrast, our approach relies on ecologically valid data based on interactions. Recent studies have highlighted the role of contextual cues and semantics in detecting deceptive or harmful discourse [13,14].

2.3. Corpora and Cross-Domain Generalization

Deception classifiers are heavily based on data realism and cross-domain transferability. The widely used Ott corpus [15], though influential, suffers from artificiality due to task framing. Barsever et al. [6] introduced the Motivated Deception Corpus, which consists of gamified statements with performance incentives. Models trained on this corpus generalized better to LIAR (53.6% vs. 47.2%). UNIDECOR [16], a unified cross-domain corpus, revealed significant performance drops (10–15%) when models trained in one domain were tested in another, highlighting domain-shift problems. Similarly, Capuozzo et al. [17] showed up to 20% accuracy loss in multilingual transfer tasks, demonstrating the limits of current generalization strategies. Taken together, these results highlight the weakness of deception models under domain shift and the importance of designing frameworks that can scale across heterogeneous sources of data.

2.4. Multimodal and Physiological Signals

While text remains central, multimodal approaches have achieved superior accuracy in naturalistic settings. Chou et al. [18] combined prosodic, acoustic, and linguistic features, reaching 85% accuracy on video-based dialogue. Krishnamurthy et al. [19] fused microexpressions, speech, and text to detect deception in courtroom recordings, outperforming text-only models by nearly 9%. These approaches illustrate the complexity of deception as a linguistic, psychological, and social phenomenon. While multimodal systems have shown strong performance, many practical settings still rely primarily on textual data. Moreover, current benchmarks often lack the social realism or fine-grained annotations necessary for deeper analysis. Our work therefore focuses on socially grounded textual deception, combining the ecological validity of interactive data with the scalability of modular computational architectures. We now describe the dataset in detail.

3. Dataset Overview

Our study relies on a recently introduced dataset by von Schenk et al. [1], specifically designed to investigate deception in text-based socially interactive settings. In contrast to traditional corpora focused on opinion spam [15] or political fact-checking [20], this dataset captures intentional first-person deceptive and truthful statements under ecologically valid conditions, providing richer linguistic and contextual cues. Unlike many existing deception corpora, which often consist of crowdsourced opinion reviews or fact-checking statements, this dataset emphasizes socially grounded communication in which participants write

about their own intentions. This design ensures higher ecological validity and provides linguistic cues that more closely resemble natural deception in everyday life.

3.1. Data Collection Protocol

A total of 986 participants were recruited through the Prolific platform to complete a two-part writing task. In the first part, participants composed a truthful statement describing their actual plans for the upcoming weekend, accompanied by a short justification to support their claim. In the second part, they produced a deceptive statement by selecting an activity they did not intend to pursue and writing a fabricated message asserting that they would.

The group of participants was diverse in terms of demographic composition (age range 18 to 65 years, balanced between genders, and including speakers from multiple English-speaking countries). This demographic variability enhances external validity of the dataset compared to corpora that often rely on more homogeneous samples. To simulate realistic social pressure, participants were informed that their deceptive messages would be evaluated by peers for credibility. Those whose lies were successfully judged as truthful received a monetary bonus (£2), introducing an incentive similar to gamified deception setups [6]. Crucially, participants were not notified in advance that they would be asked to lie, reducing strategic pre-planning and encouraging spontaneous language use.

3.2. Data Validation and Cleaning

All submissions underwent a two-stage validation process. First, automatic filtering removed entries shorter than 150 characters. Second, trained research assistants manually reviewed submissions for clarity, internal consistency between the truthful statement and its justification, and adherence to instructions. Participants whose submissions failed to meet these criteria were excluded from the final dataset.

After filtering, the corpus contained 768 participants, each contributing one truthful statement (with justification) and one deceptive statement. This resulted in a balanced dataset of 1536 natural language messages, annotated both for actual veracity and for perceived truthfulness as judged by peers.

3.3. Annotation Schema

Each entry includes metadata supporting nuanced analyses of deception.

- *true_statement*: binary indicator of whether the message reflects the author's real intent,
- *justification_text*: rationale provided only for truthful statements,
- *judged_truthful*: binary label reflecting whether a peer evaluator believed the statement to be true.

This dual labeling, which captures both ground truth and subjective interpretation, enables research not only on how deception is generated, but also on how it is perceived by human evaluators, aligning with recent calls for richer deception datasets [11]. This structure distinguishes the dataset from widely used benchmarks such as LIAR or Ott, which provide only binary ground truth labels without capturing the perception of deception by human judges.

Table 1 summarizes the dataset following validation, highlighting its balanced structure, inclusion of justifications, and availability of dual veracity annotations. These properties make it particularly well suited for studying deception in linguistically rich, socially grounded contexts. Compared with prior corpora such as Ott, LIAR, or UNIDECOR, the present data set provides a unique combination of ecological realism, demographic diversity, and dual labeling, offering stronger foundations for cross-domain generalization and interpretability studies.

Table 1. Key attributes of the deception dataset.

Attribute	Value
Total participants (initial)	986
Final participant count (post-filtering)	768
Total statements	1536
Truthful statements	768
Deceptive statements	768
Minimum statement length	150 characters
Justification included	Yes (truthful only)
Veracity label (true_statement)	Yes
Perceived truth label (judged_truthful)	Yes
Demographic diversity	Age 18–65, mixed gender, multiple countries

4. Methodology: Self-Verification Loop for Lie Detection

The pipeline is structured as a modular architecture, where each reasoning stage can be implemented, replaced, or scaled independently. This design supports distributed deployment in large-scale data processing environments. Our methodology combines two complementary components: (1) a sentence decomposition stage that improves the interpretability of messages, and (2) a self-feedback refinement process using a pre-trained large language model (LLM), specifically Gemma and LLaMA, to detect deception in dialogues (See Figure 1). Each stage operates with linear time complexity in the number of input sentences, allowing the framework to handle large datasets efficiently. The modular nature facilitates parallelization and integration into big data infrastructures. The rationale is that deception often relies on subtle inconsistencies that are easily missed by monolithic classifiers. By explicitly decomposing utterances and then applying iterative self-reflection, the approach mimics human-like reasoning, where initial judgments are revised after critical evaluation. This two-stage approach increases both interpretability and robustness in settings where labeled data are scarce.

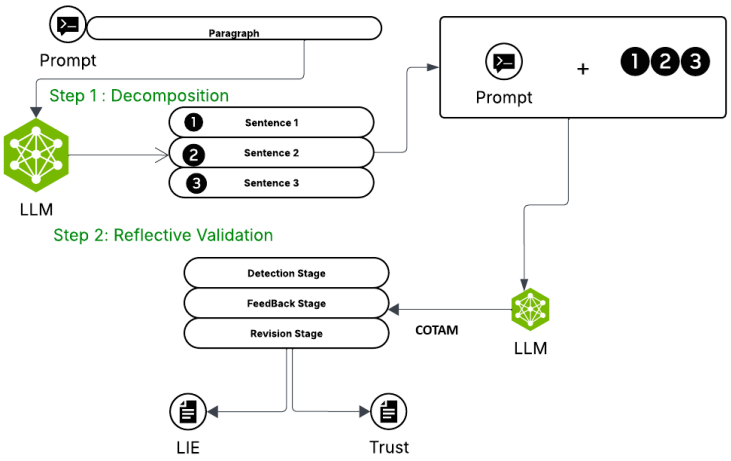

Figure 1. An overview of our method (1) Decomposition and (2) Reflective validation.

4.1. Prompt Templates

For transparency and reproducibility, we report here the complete prompts used in our experiments.

Prompt 1—Baseline Test

```
You  are a lie detection assistant. Your task is to determine whether a statement
is a lie (label = 1) or a truth (label = 0).

Base your decision on the following criteria:

1. Internal Contradiction---Does the statement contradict itself?
2. Low Plausibility---Is the situation unlikely or exaggerated?
3. Ambiguity---Is the statement vague, confusing, or evasive?
4. Inconsistency with Common Knowledge---Does it go against widely accepted facts?
5. Over-Embellishment---Does it seem to try too hard to impress or justify?
6. Implausible Denial---Is blame shifted in an unconvincing way?

Indicators of truthful statements include:
- Clear, coherent, and specific details
- Neutral or factual tone
- Alignment with general world knowledge
- No excessive persuasion or justification attempts

Return only the final decision using the following format:
<label>1</label> for lie, <label>0</label> for truth.

Example:
Statement: "I never eat sugar, but I had cake and cookies last night."
Your answer: <label>1</label>

Statement: "Yesterday I worked from 9 a.m. to 5 p.m. and took a 30-min lunch break."
Your answer: <label>0</label>

Now analyze the following statement and return only the label:
{text}
```

Prompt 2—Feedback with Explanation

```
You are a lie detection assistant. Your task is to evaluate a statement and determine whether it
is a lie (label = 1) or a truth (label = 0). Provide a clear justification for your
judgment based on the criteria˜below.

Use the following guidelines to assess the statement:

1. Internal Contradiction---Does the statement contradict itself?
2. Low Plausibility---Is the situation unlikely or exaggerated?
3. Ambiguity---Is the statement vague, confusing, or evasive?
4. Inconsistency with Common Knowledge---Does it go against well-known facts?
5. Over-Embellishment---Does it try too hard to sound impressive or detailed?
6. Implausible Denial---Does it deflect responsibility in an unconvincing way?

Indicators of a truthful statement include:
- Clear and consistent content
- Neutral and factual tone
- Plausible events or details
- Alignment with common knowledge
- No excessive˜persuasion

Your response must follow this format:

<explanation>
[Your explanation here: Explain why the statement is a lie or truth based on the criteria above.]
</explanation>
<label>[0 or 1]</label>

Examples:

Statement: "I never eat sugar, but I had cake and cookies last night."

<explanation>
This statement contradicts itself. Claiming to never eat sugar is
inconsistent with admitting to having cake and cookies,
which clearly contain sugar. This contradiction suggests dishonesty.
</explanation>
<label>1</label>

Statement: "Yesterday I worked from 9 a.m. to 5 p.m. and took a 30-min lunch break."
```

```
<explanation>
The statement is clear, plausible, and contains specific time-based details.
There is no sign of contradiction or exaggeration.
It aligns with common daily routines and seems factually neutral.
</explanation>
<label>0</label>

Now analyze the following statement and provide your reasoning and label:
{text}
```

Prompt 3—Reviewer Prompt

```
You are a lie detection reviewer. Your job is to critique the explanation
and label given by a previous~assistant.

Your goals are:
1. Evaluate the quality and logic of the explanation.
2. Identify if the decision (label) was correct or flawed.
3. If needed, revise the label and justify your reasoning.

You must focus on the following evaluation criteria:
- Did the explanation address key deception indicators?
- Was the reasoning coherent, factual, and complete?
- Is the assigned label (0 or 1) consistent with the explanation?

Here is the original statement and assistant's response:

Statement: "{text}"

Initial Explanation (Step 1):
{explanation}

Initial Label: {initial_label}

Now write your feedback and revised judgment, using the following format:

<feedback>
[Write your critique here. Mention strengths or weaknesses in the explanation.
Suggest what should be improved or why you agree/disagree with the initial reasoning.]
</feedback>

<label>[1 or 0]</label>
```

Prompt 4—Final Decision Prompt

```
You are a lie detection assistant reviewing previous analyses to make a final decision.

Below is a statement (or paragraph), along with:
- An initial explanation and label from another assistant
- A reviewer's critical feedback and revised suggestion

Your task is to:
1. Carefully consider the original statement, the initial reasoning, and the reviewer's feedback
2. Decide whether the statement is a lie (label = 1) or a truth (label = 0)
3. Output only the final label inside the <label></label> tag

Do not repeat the explanation or feedback. Respond only with the final decision.
Statement: {text}

Initial explanation with label {initial_label}:
{explanation}

Reviewer feedback with revised label {revised_label}:
{feedback}
```

4.2. Message Decomposition

To enhance the granularity of the analysis, we decompose each full message into smaller, semantically independent phrases. This approach is inspired by the compositional translation method introduced in [21], where complex sentences are split into simpler, minimal propositions that retain lexical overlap with the original text.

We use a divide prompt with in context examples adapted from the MinWikiSplit corpus [22], which allows us to generate decompositions automatically via prompting. Each resulting phrase expresses a single proposition, improving the precision of the prediction of downstream deception. The number of phrases generated is not fixed, but is determined instead by the structure and complexity of the input sentence. This step is important because deceptive language often embeds misleading fragments within otherwise truthful discourse. Decomposition allows the system to isolate and evaluate each fragment separately, making inconsistencies more apparent to the LLM.

4.3. Lie Detection Strategy

Following the decomposition phase, we use a self-refinement framework based on large-language models to detect deception. This is fully detailed in Section 5, where we introduce our validation framework called COTAM (Chain-of-Thought Assisted Modification), combining prediction, feedback, and revision stages to iteratively refine the LLM's decisions. Unlike previous methods that use single-pass classification, our strategy emphasizes iterative reasoning. Each prediction is subject to critique and revision, which reduces the risk of overfitting to superficial lexical cues and increases the connection between model decisions and linguistic evidence.

5. Validation Strategies for LLM-Augmented Misinformation Data

5.1. LLM-As-a-Judge: Prompted Consistency Checking

Input: Sentence S **Output:** Binary label (0 = truth, 1 = lie) and justification

Step 1: Criteria for lie detection

- *Internal contradiction*—sentence contradicts itself
- *Low plausibility*—unlikely or exaggerated
- *Ambiguity*—vague or evasive content
- *Conflict with common knowledge*—contradicts known facts
- *Overstatement/embellishment*—trying too hard to impress
- *Unconvincing denial*—implausible deflection

Step 2: Indicators of truth

- Clear and coherent details
- Neutral or factual tone
- Aligns with general knowledge
- No excessive persuasion

Step 3: Analyze sentence

1. Examine S according to the criteria
2. Generate reasoning chain (CoT) explaining your judgment

Step 4: Preliminary output

- Provide binary label (0 or 1)
- Attach explanation justifying the decision

Step 5 (Optional): Rewriting

- If S is false but intended to be truthful:
- Prompt: "*Rewrite this sentence so it is logically and factually accurate.*"

5.2. COTAM: Chain-of-Thought Assisted Modification for Lie Detection

Goal: Detect deception in messages without fine-tuning or labeled supervision using a self-feedback pipeline.

Pipeline: COTAM applies LLM reasoning sequentially over each sentence or paragraph in three stages: detection, critique, revision.

1. **Detection Stage (Initial Judgment)**
   - Input: sentence S
   - Task: assign binary label (0 = truth, 1 = lie) with justification
   - Method: CoT-based prompt from LLM-as-a-Judge (Section 4.1)
   - Output: preliminary prediction + explanation
2. **Feedback Stage (Self-Critique)**
   - Task: critically review previous explanation and label
   - No access to ground truth
   - Output: revised judgment and rationale if needed
   - Example Prompt:

     "You are a lie detection critic. Review the reasoning and decision of another assistant.
     Sentence: {text}
     Initial Explanation: {explanation}
     Initial Label: {first answer}
     Assess whether the reasoning is sound and whether the label is correct. Revise the label and explain your reasoning if needed."
3. **Revision Stage (Final Decision)**
   - Task: synthesize original explanation + critique to finalize label
   - Example Prompt:

     "Considering the above critiques, what messages do you still consider lies? Justify your final decision."
   - Output: final label (ultimate deception classification)

Notes: COTAM leverages self-refinement paradigms [23], decomposes messages into individual sentences, and applies step-by-step reasoning to enhance interpretability and robustness without labeled data or fine-tuning. This framework is therefore not only a classifier but also a reasoning protocol, bridging the gap between raw prediction and explainability.

6. Experiments and Results

6.1. Experimental Results

We evaluated our framework on the dataset described in Section 3, comparing three major configurations: a one-shot baseline inference, a multistep LLM feedback pipeline (COTAM), and a decomposition-enhanced variation. In addition, we explore a hybrid system that combines multiple models in various reasoning stages. The overall performance summary is reported in Table 2.

Table 2. Performance Summary Across Models and Reasoning Strategies (including Precision, Recall, and Accuracy).

Model	Method	Dataset	F1-Lie	Macro-F1	Pr.	Rec.	Acc.
Gemma-3-27B	baseline	lie_detection	0.1927	0.4274	0.5717	0.5235	0.4797
Gemma-3-27B	feedback	lie_detection	0.6152	0.4344	0.4867	0.4922	0.4816
Gemma-3-27B	baseline	decomposed_v2	0.5750	0.4400	0.5690	0.5235	0.5190
Gemma-3-27B	feedback	decomposed_v2	0.5750	0.4400	0.4642	0.4725	0.5717
LLaMA-3-8B	baseline	lie_detection	0.3461	0.4681	0.4950	0.4961	0.4867
LLaMA-3-8B	feedback	lie_detection	0.2551	0.4372	0.4933	0.4961	0.5414
LLaMA-3-8B	baseline	decomposed_v2	0.2889	0.4505	0.5173	0.5098	0.4642
LLaMA-3-8B	feedback	decomposed_v2	0.2889	0.4505	0.4970	0.4980	0.5690
Mistral-7B	baseline	lie_detection	0.2561	0.4517	0.5440	0.5216	0.4642
Mistral-7B	feedback	lie_detection	0.4941	0.4941	0.4941	0.4941	0.4950
Mistral-7B	baseline	decomposed_v2	0.4980	0.5058	0.6036	0.5392	0.4933
Mistral-7B	feedback	decomposed_v2	0.4980	0.5058	0.5059	0.5059	0.5150
Combined	feedback	decomposed_v2	0.4434	0.5089	0.5190	0.5176	0.4907

6.1.1. Baseline Performance

In the one-shot inference scenario, the LLMs received minimal context, one truthful and one deceptive example, and were tasked with binary classification. Gemma-3-27B achieved an F1_lie of only 0.19, compared to 0.35 for LLaMA-3-8B and 0.26 for Mistral-7B. These results indicate that the three models tend to adopt conservative behavior, favoring the prediction of 'truthful' rather than 'lie'. Among them, LLaMA-3-8B stands out as the most reliable in this baseline setting, outperforming Gemma by roughly 16 percentage points and Mistral by about 9 points on F1_lie. Gemma's particularly low performance highlights that a single forward pass without feedback is insufficient for nuanced deception classification.

Although the main objective of this paper is to enhance reasoning methods for LLMs, we also report results from a strong transformer-based reference model (mDeBERTa-v3-base), widely used in deception detection, to provide a comparative baseline.

As shown in Table 3, the transformer baseline achieves an accuracy of 56.5% and a Macro-F1 of 0.48, placing it slightly above the three LLMs in terms of balanced performance. However, the difference is not dramatic: for instance, LLaMA-3-8B reaches a Macro-F1 of 0.47, only 1.4 points lower.

Table 3. Baseline performance of LLMs and transformer model (lie_detection dataset).

Model	F1-Lie	Macro-F1	Precision	Recall	Accuracy
Gemma-3-27B	0.1927	0.4274	0.5717	0.5235	0.4797
LLaMA-3-8B	0.3461	0.4681	0.4950	0.4961	0.4867
Mistral-7B	0.2561	0.4517	0.5440	0.5216	0.4642
mDeBERTa-v3-base	0.2283	0.4825	0.5551	0.4868	0.5647

Looking more closely, the LLM baselines generally exhibit higher precision than recall. For example, Gemma-3-27B reached a precision of 57.2% versus a recall of 52.3%, and Mistral-7B obtained 54.4% precision versus 52.2% recall. This imbalance confirms their conservative tendency: they predict "truthful" more often than "lie," which drives down their F1-lie despite moderate overall precision. By contrast, mDeBERTa-v3-base produces more balanced predictions, with recall (48.7%) closer to its overall accuracy, though it still struggles to reliably identify deceptive cases. To assess whether the observed improvements were statistically significant, we conducted paired t-tests comparing the performance of each model configuration. Results show that the gains from COTAM and

decomposition strategies were significant ($p < 0.05$) for Gemma and Mistral, confirming that iterative reasoning and sentence-level decomposition meaningfully enhance deception detection. Unlike LLMs, transformer-based models such as mDeBERTa-v3-base cannot leverage iterative feedback or self-reflection; their performance depends solely on single-pass inference.

6.1.2. LLM Feedback Framework

Introducing the three-step LLM feedback pipeline significantly improved model performance. Gemma benefited the most, with F1-lie increasing from 0.1927 to 0.6152 when operating in the full message context, representing a relative improvement of 141% compared to the baseline. In contrast, LLaMA experienced a significant decrease from 0.3461 to 0.2551, suggesting that reflective prompting did not substantially enhance its initial predictions. Mistral outperformed both models in this configuration, achieving an F1-lie of 0.4941, representing a 93% improvement over its baseline. In terms of accuracy, improvements were more moderate: Gemma rose slightly from 0.4797 to 0.4816, LLaMA from 0.4867 to 0.5414, and Mistral from 0.4642 to 0.4950. This shows that while F1-lie can increase substantially (especially for Gemma), accuracy gains are smaller, reflecting the trade-off between detecting more lies and occasionally misclassifying truthful statements. Qualitatively, the feedback stage enabled models to flag deceptive cues such as vague justifications (e.g., *"I will definitely study hard this weekend"*), which lack concrete details and were often overlooked in the baseline setting. Feedback reduces precision but increases balance between precision and recall (e.g., Gemma drops from 0.5717 precision to 0.4867 but recall stabilizes around 0.4922), indicating that reflective prompting encourages models to take more risks in flagging lies, improving F1-lie at the expense of precision.

6.1.3. COTAM Self-Feedback

Applying the self-feedback loop without decomposition further enhanced performance. Gemma's F1-lie rose to 0.6152, which represents a 27% increase compared to the standard feedback approach and more than 220% compared to the baseline, reflecting the strong impact of iterative self-reflection. LLaMA slightly regressed to 0.2551, possibly due to overconfidence in its initial responses, while Mistral maintained balanced performance at 0.4941, indicating consistent gains from feedback-based reasoning. This stage also brought moderate accuracy improvements: Gemma reached 0.5717 in the decomposed feedback setting, compared to 0.4797 in baseline, while Mistral and LLaMA stabilized around 0.4950 and 0.5414 respectively. It was particularly effective in detecting contradictions across multiple propositions. For instance, the fabricated message *"I will go hiking on Saturday and stay home all weekend"* was correctly flagged as deceptive after COTAM's critique cycle, while both baseline and single-pass feedback models failed.

6.1.4. Sentence Decomposition

Incorporating sentence-level decomposition (decomposed_v2) to produce atomic propositions further stabilized predictions across both LLMs and transformer models. The decomposition strategy reduced input complexity, aiding local reasoning and mitigating potential hallucinations during iterative feedback.

For LLMs: Gemma achieved an F1-lie of 0.5750, slightly lower than the COTAM-only version but still almost triple its baseline score. LLaMA improved modestly to 0.2889 (+13%) and Mistral reached 0.4980 (+0.8%). A typical gain came from cases where participants embedded partial truths into deceptive messages. For example, the statement *"I will go shopping with friends and also attend a yoga retreat"*, where only the second clause was fabricated, was successfully decomposed into atomic propositions, allowing the LLMs to identify the deceptive fragment in isolation.

For mdeberta_v3_base: the model shows consistent improvements with decomposition. On the original lie_detection dataset, accuracy = 0.5647, F1 = 0.4868; with lie_detection_decomposed, accuracy = 0.5667, F1 = 0.4938; and with lie_detection_decomp._v2, accuracy = 0.6275, F1 = 0.5853. This demonstrates that sentence-level decomposition (v2) significantly boosts performance, particularly for models that benefit from shorter, atomic inputs, improving both F1 and accuracy.

Overall, across both LLMs andmdeberta_v3_base, sentence decomposition enables more precise detection of deceptive content by isolating atomic propositions and enhancing local reasoning. With decomposition, precision improves markedly (Mistral: 0.6036) while recall also rises (0.5392), showing that atomic proposition breakdown enhances both the sensitivity and reliability of lie detection. Gemma, however, suffers a precision drop in feedback mode (0.4642), highlighting that decomposition benefits are model-dependent.

6.1.5. Hybrid Chain-of-Models

Finally, the hybrid configuration distributed responsibilities between models: Mistral performed the initial test and justification, LLaMA handled critique, and Gemma provided the final decision. This approach produced an F1-lie of 0.4434 and a Macro-F1 of 0.5089. While it did not achieve the highest class-specific F1, it offered the most balanced and interpretable predictions, suggesting that inter-model validation can mitigate individual biases and enhance overall robustness. This hybrid strategy illustrates a promising direction: combining complementary strengths of LLMs to create an ensemble pipeline where disagreements lead to more cautious, evidence-based final decisions. The hybrid system maintains a balanced trade-off, with precision (0.5190) and recall (0.5176) closely aligned, confirming that distributing reasoning tasks across LLMs stabilizes performance and reduces bias toward either precision or recall.

7. Computational Cost Analysis

We evaluated inference times under two conditions: (i) simple prediction (binary label, <8 tokens), and (ii) reasoning with feedback (three consecutive queries: explanation, revision after feedback, final decision). In the simple setting, the average execution time per query was 0.173 s for Gemma-27B, 0.068 s for LLaMA-8B, and 0.073 s for Mistral-7B. Importantly, decomposed statements did not noticeably increase inference time: the average latency differences across models were marginal (<0.02 s). In the feedback setting, the cost increases substantially due to longer generations. Gemma-27B required 7.7 s per input on average (1530 queries for 510 statements), LLaMA-8B required 2.8 s, and Mistral-7B 1.9 (see Table 4). Yet, the difference between original and decomposed statements remained negligible, indicating that decomposition improves reasoning robustness without incurring additional computational cost. Overall, although reasoning with feedback is more expensive than direct predictions, the total runtime remains manageable on a single NVIDIA RTX 5090 GPU. We do not take into account the cost of decomposition since it is stored in separate files.

Table 4. Average inference time per query (in seconds) for simple and feedback settings, with and without statement decomposition (510 inputs).

Model	Setting	Original	Decomposed	δ
Gemma-27B	Simple	0.173	0.175	+0.002
	Feedback	7.717	7.72	$\approx$0
LLaMA-8B	Simple	0.068	0.081	+0.013
	Feedback	2.811	2.81	$\approx$0
Mistral-7B	Simple	0.073	0.069	$-$0.004
	Feedback	1.875	1.87	$\approx$0

8. Discussion

8.1. Feedback and Decomposition Synergy

Our results reinforce the importance of multi-layered reasoning for lie detection. Self-feedback markedly improves sensitivity to deception, especially for Gemma, which triples its F1-lie when using COTAM. Decomposition adds a second layer of support, allowing LLMs to focus on smaller semantic units, which improves consistency and clarity of classification. This synergy shows that deception is rarely exposed by a single lexical cue but rather emerges from contradictions across fragments. By forcing models to alternate between microlevel decomposition and macrolevel feedback, we replicate the human strategy of zooming in and out of discourse, yielding greater robustness.

Interestingly, Mistral consistently performs robustly, while Gemma exhibits higher variance across settings but responds best to structured guidance. LLaMA performs poorly in feedback-based settings, but demonstrates potential in isolated or ensemble roles. This variability across models also highlights the importance of architecture-specific reasoning styles, suggesting that future pipelines should dynamically allocate roles based on each model's inductive bias rather than treating all LLMs as interchangeable.

8.2. Challenges and Ambiguity

Despite measurable gains, deception detection remains a challenge. Many truthful and deceptive statements differ subtly and require pragmatic or world knowledge to disambiguate. For instance, deceptive statements often mimic sincerity by using specificity or a neutral tone, which can mislead both humans and models. For example, messages such as "I will spend Saturday working on my thesis and then relaxing with friends" are extremely difficult to classify, since they combine plausible self-disclosure with socially acceptable behavior, blurring the line between fabricated and genuine intent.

Additionally, sentence-level classification overlooks strategic intent, sarcasm, or high-level discourse features. These limitations suggest that future work should incorporate cross-sentence coherence, speaker history, and possibly external factual verification modules. Another challenge is dataset bias: although socially grounded, our corpus is still restricted to written, short-term deceptive statements, which may not fully reflect long-term, high-stakes deception in domains like finance or politics. This gap must be addressed before real-world deployment.

8.3. Modular Reasoning Across Models

The "combined" configuration demonstrates that distributing tasks across multiple LLMs can yield more robust decisions. This strategy mitigates individual model weaknesses and introduces meta-validation steps. In real-world deployments, such modular pipelines may enhance robustness, interpretability, and safety, particularly for high-stakes use cases (e.g., legal, hiring, or moderation systems). In practice, this means that a re-

cruitment platform could integrate a "lie detection module" where one LLM evaluates candidate claims, another critiques the judgment, and a third finalizes the decision. Such checks-and-balances mimic institutional oversight mechanisms, offering both accuracy and accountability.

Beyond performance gains in deception detection, the proposed approach demonstrates computational robustness and potential for integration into real-world information systems where large text streams must be analyzed continuously. By structuring deception detection as a multi-stage computational pipeline, our method can be adapted to other NLP tasks that require modular reasoning and scalable deployment.

9. Conclusions

In this work, we introduced a multistep framework for textual lie detection that integrates semantic decomposition with self-reflective reasoning through the novel Chain-of-Thought Assisted Modification (COTAM) process. Experimental results with Gemma, LLaMA, and Mistral demonstrate that structured prompting combined with modular reasoning substantially improves classification performance, particularly when messages are analyzed in decomposed formats. Furthermore, chaining models in a hybrid architecture yields consistent and interpretable predictions, indicating the potential of such pipelines for advanced deception-aware NLP systems. The modular design supports flexible extensions, including integration with external knowledge bases for fact-checking and incorporation of multimodal modules for analyzing voice and facial expressions. Future research directions include incorporating conversational context, grounding models in external knowledge, and combining textual analysis with behavioral or multimodal cues to enhance detection reliability. Longitudinal tracking of deceptive intent is another promising avenue, enabling systems to capture patterns of inconsistency across multiple statements over time. Beyond its technical contributions, this framework addresses broader concerns about trust, transparency, and safety in AI-mediated communication. By enhancing the reliability of NLP systems for detecting deceptive content, this work supports the development of robust digital infrastructures aligned with Sustainable Development Goals (SDGs), particularly in areas such as digital literacy, institutional accountability, and responsible AI deployment. Overall, this study contributes both to improved deception detection and to the advancement of computational methods for reasoning pipelines, scalable NLP architectures, and data-driven cognitive computing systems. Future work will focus on distributed deployments and integration within enterprise-scale information systems.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in: https://osf.io/eb59s/metadata/osf (accessed on 11 May 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. von Schenk, A.; Klockmann, V.; Bonnefon, J.F.; Rahwan, I.; Köbis, N. Lie detection algorithms disrupt the social dynamics of accusation behavior. *iScience* **2024**, *27*, 110201. [CrossRef] [PubMed]
2. Newman, M.L.; Pennebaker, J.W.; Berry, D.S.; Richards, J.M. Lying words: Predicting deception from linguistic styles. *Personal. Soc. Psychol. Bull.* **2003**, *29*, 665–675. [CrossRef] [PubMed]
3. Vrij, A. *Detecting Lies and Deceit: Pitfalls and Opportunities*; John Wiley & Sons: Hoboken, NJ, USA, 2008.

4. Hancock, J.T.; Toma, C.L.; Ellison, N.B. The truth about lying in online dating profiles. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Florence Italy, 5–10 April 2008; pp. 449–452.
5. Levitan, S.I.; Maredia, A.; Hirschberg, J. Identifying indicators of veracity in text-based deception using linguistic and psycholinguistic features. In Proceedings of the Second Workshop on Computational Approaches to Deception Detection, San Diego, CA, USA, 17 June 2016; pp. 25–34.
6. Barsever, D.; Steyvers, M.; Neftci, E. Building and benchmarking the Motivated Deception Corpus: Improving the quality of deceptive text through gaming. *Behav. Res. Methods* **2023**, *55*, 4478–4488. [CrossRef] [PubMed]
7. Ilias, L.; Askounis, D. Multitask learning for recognizing stress and depression in social media. *Online Soc. Netw. Media* **2023**, *37–38*, 100270. [CrossRef]
8. Toma, C.L.; Hancock, J.T. Lies in online dating profiles. *J. Soc. Pers. Relationsh.* **2010**, *27*, 749–769.
9. Fornaciari, T.; Poesio, M. Automatic deception detection in Italian court cases. In *Artificial Intelligence and Law*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 21, pp. 303–340.
10. Niculae, V.; Danescu-Niculescu-Mizil, C. Linguistic harbingers of betrayal: A case study on an online strategy game. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), Beijing, China, 26–31 July 2015; pp. 1793–1803.
11. Peskov, D.; Yu, M.; Zhang, J.; Danescu-Niculescu-Mizil, C.; Callison-Burch, C. It takes two to lie: One to lie, and one to listen. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 5–10 July 2020; pp. 3811–3825.
12. Baravkar, P.; Shrisha, S.; Jaya, B.K. Deception detection in conversations using the proximity of linguistic markers. *Knowl.-Based Syst.* **2023**, *267*, 110422. [CrossRef]
13. Belbachir, F.; Roustan, T.; Soukane, A. Detecting Online Sexism: Integrating Sentiment Analysis with Contextual Language Models. *AI* **2024**, *5*, 2852–2863. [CrossRef]
14. Bevendorff, J. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, Style Change Detection, and Trigger Detection. In Proceedings of the CLEF 2022, Bologna, Italy, 5–8 September 2022.
15. Ott, M.; Choi, Y.; Cardie, C.; Hancock, J.T. Finding deceptive opinion spam by any stretch of the imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), Portland, OR, USA, 19–24 June 2011; pp. 309–319.
16. Velutharambath, A.; Klinger, R. UNIDECOR: A unified deception corpus for cross-corpus deception detection. In Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis (WASSA), Toronto, ON, Canada, 14 July 2023; pp. 39–51. [CrossRef]
17. Capuozzo, P.; Lauriola, I.; Strapparava, C.; Aiolli, F.; Sartori, G. DecOp: A multilingual and multi-domain corpus for detecting deception in typed text. In Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC), Marseille, France, 11–16 May 2020; pp. 1394–1401.
18. Chou, C.C.; Liu, P.Y.; Lee, S.Z. Automatic deception detection using multiple speech and language communicative descriptors in dialogs. *Apsipa Trans. Signal Inf. Process.* **2022**, *11*, e31. [CrossRef]
19. Krishnamurthy, G.; Majumder, N.; Poria, S.; Cambria, E. A deep learning approach for multimodal deception detection. *Expert Syst. Appl.* **2018**, *124*, 56–68. [CrossRef]
20. Perez-Rosas, V.; Kleinberg, B.; Lefevre, A.; Mihalcea, R. Automatic detection of fake news. In Proceedings of the 27th International Conference on Computational Linguistics (COLING), Santa Fe, NM, USA, 20–26 August 2017; pp. 3391–3401.
21. Zebaze, A.; Sagot, B.; Bawden, R. Compositional Translation: A Novel LLM-based Approach for Low-resource Machine Translation. *arxiv* **2025**, arXiv:2503.04554. [CrossRef]
22. Niklaus, C.; Freitas, A.; Handschuh, S. MinWikiSplit: A Sentence Splitting Corpus with Minimal Propositions. In Proceedings of the 12th International Conference on Natural Language Generation, Tokyo, Japan, 29 October–1 November 2019; van Deemter, K., Lin, C., Takamura, H., Eds.; pp. 118–123. [CrossRef]
23. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Welleck, S.; Majumder, B.P.; Gupta, S.; Yazdanbakhsh, A.; Clark, P. Self-Refine: iterative refinement with self-feedback. In Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23), Orleans, LA, USA, 10–16 December 2023; pp. 2019–2079.