Team JRC-GENESIS at PAN 2026: SYN^2 – Hybrid Watermarking via Keyed SYNonym Substitution and SYNtactic Signals

Notebook for the PAN Lab at CLEF 2026

Elisa Di Nuovo[1,*,‡], Davide Colla[1,‡], Quang Luong[1] and Simone Ceppi[1]

[1]European Commission, Joint Research Centre, Via Enrico Fermi, 2749 – 21027 Ispra (VA), Italy

Abstract

We present a hybrid linguistic watermarking system for the PAN at CLEF 2026 Text Watermarking shared task. Our approach embeds watermarks through two principal channels operating at different linguistic levels: (1) a *keyed channel* that performs keyed synonym substitution using WordNet with corpus-aware filtering, and contraction transformations, and (2) a *key-independent channel* that applies structural transformations including domain-specific named entity expansion, verb-conjunction semicolon substitution, participial clause expansion, and active-to-passive voice conversion. Detection aggregates evidence from all channels via a combined log-likelihood ratio test. We submit two system versions: v1 uses WordNet with Wu–Palmer similarity for synonym selection and a simple LLR-based detection; v2 extends v1 with BERTScore-based synonym selection, additional attack-resistant syntactic signals, and a gating mechanism. We evaluated both versions on three validation sets with multi-level attacked and non-attacked watermarked and original texts. Code is available at: https://github.com/davidecolla/pan-clef-2026-wt-jrc-genesis. These versions, v1 and v2, achieved on the official test set a TWF of 0.686 and 0.761, respectively.

Keywords

text watermarking, corpus linguistics, synonym substitution, syntactic transforms, log-likelihood ratio test

1. Introduction

Text watermarking embeds imperceptible signals into natural language text that can later be detected to verify provenance or authorship. The PAN at CLEF 2026 shared task on Text Watermarking [1, 2] challenges participants to develop watermarking systems that are robust to different types of attacks while preserving text quality, evaluated on the TIRA platform [3].

This shared task addresses a fundamentally different problem from recent work on watermarking Large Language Model (LLM) outputs [4]. LLM watermarking operates *during generation*: the model's token sampling is biased toward a "green list" of tokens determined by a hash of preceding context, creating a statistical signature detectable in the output. In contrast, the PAN task requires *post-hoc watermarking* of the existing human-written text. This setting could be addressed using a LLM which would paraphrase the original text and watermark the text at the same time, while maintaining text quality. We adopted a lightweight corpus-driven approach, which offers an efficient solution with modest computational requirements. We start by understanding the linguistic characteristics of the target domain to identify modifications that will (1) maintain original text quality (measured via BERTScore, as in the official evaluation), (2) create detectable signal, and (3) remain robust against various attacks.

We propose a multi-channel approach that distributes the watermark signal across multiple independent linguistic dimensions. The key insight is that an attacker who paraphrases text may destroy lexical

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

*Corresponding author.

‡These authors contributed equally to this work.

elisa.di-nuovo@ec.europa.eu (E. Di Nuovo); davide.colla@ec.europa.eu (D. Colla); quang.luong@ec.europa.eu (Q. Luong); simone.ceppi@ec.europa.eu (S. Ceppi)

0000-0002-4814-982X (E. Di Nuovo); 0009-0004-1542-0048 (D. Colla); 0000-0001-6505-7313 (Q. Luong); 0000-0002-0457-1340 (S. Ceppi)

signals (by replacing words) but is unlikely to simultaneously undo syntactic restructuring, contraction patterns, and domain-specific term choices. By combining evidence from orthogonal channels, our detector achieves robustness that no single channel could provide alone (Section 5).

Our system uses the NLTK library to split documents into sentences (to use the same sentence splitting used for the official evaluation) and spaCy for linguistic text processing. Both versions use WordNet for synonym candidate generation. We submit two versions: ***(i)*** **v1** – Three-channel system (WordNet lexical + contraction + syntactic) with combined Log-Likelihood Ratio (LLR) detection and Wu–Palmer similarity for synonym selection. ***(ii)*** **v2** – Extends v1 with BERTScore-based synonym selection and stricter syntactic transformation handling. For increased watermarking recognition it adds domain-specific institution signals, attack-resistant syntactic features in the detector, and a gating mechanism against false positives.

2. Related Work

Text watermarking has evolved from early format-based manipulations to sophisticated linguistic transformations and, more recently, to statistical distortions within generative AI architectures. We classify previous work into four areas relevant to our system.

Format-based transformations. Early methods embed information by editing document layout: line-shift and word-shift coding [5], subsequently refined through systematic comparison [6, 7] and improved spacing schemes [8, 9, 10, 11]. However, these methods are limited to image-formatted text. Unicode-based methods such as UniSpach [12] and homoglyph substitution [13] extend to raw text but remain vulnerable to canonicalization attacks.

Linguistic watermarking. Linguistic watermarking leverages the "hiding virtues of ambiguity" [14] inherent in natural language. *Lexical substitution* replaces content words with synonyms from a lexical database like WordNet [15], derived via a secret key. Atallah et al. [16] established the foundational framework; Topkara et al. [14] demonstrated WordNet-based synonym substitution with resilience analysis. The primary challenge is avoiding "semantic drops" where substitutions break contextual coherence [17, 18]; we address this through corpus-aware filtering and Wu-Palmer similarity thresholds [19]. *Syntactic transformations* operate on sentence structure: Topkara et al. [20] proposed transformations on syntactic trees (e.g., voice conversion, adjunct reordering), while Meral et al. [18] demonstrated that structural changes survive edits that destroy lexical signals. Our system combines both traditions with corpus-derived statistical baselines.

Watermarking LLM outputs. The foundational "green-red list" algorithm [21] partitions the vocabulary at each generation step based on a hash of preceding tokens and a secret key, biasing sampling toward "green" tokens. Detection uses a z-score test or LLR over green-token frequency. Christ et al. [22] established theoretical foundations for undetectable watermarks, and SynthID-Text [23] demonstrated a production-scale deployment. Kirchenbauer et al. [21] further quantified robustness under paraphrase budgets, while Zhao et al. [24] gave provable guarantees under bounded edit-distance attacks. These methods are effective on freshly generated text but inapplicable to existing documents.

Attacks and robustness. Krishna et al. [25] showed that round-trip paraphrasing substantially degrades detection rates, motivating retrieval-based defences. Sadasivan et al. [26] argued that as language models approach human-level fluency, reliable detection becomes information-theoretically difficult.

Our system inherits its building blocks from the linguistic watermarking literature [16, 14, 18], but combines them in a novel framework evaluated against classical and modern attacks. Three design choices distinguish our contribution. First, we treat lexical, surface, and syntactic dimensions as orthogonal channels aggregated via a unified LLR test (Section 4). Second, we ground lexical and syntactic choices in corpus-driven information. Third, motivated by Krishna et al. [25] and Kirchenbauer et al. [21], v2 engineers attack-resistant syntactic signals in the detector to reinforce the watermarker against paraphrasing attacks.

3. Corpus Analysis

Training set analysis. We analysed the 300 training documents using ad hoc scripts and Sketch Engine [27].[1] Key findings:

- **Lexical statistics**: Mean document length of 432 words (std=197), vocabulary of 7,793 types over 115,092 tokens, 40% hapax legomena and high lexical diversity (MATTR=0.67 [28]).
- **Syntactic baselines** (NLTK sentence splitting, spaCy `en_core_web_sm`, document-level averaging): Semicolons/sentence: 0.023; passive ratio: 20%; ACL ratio: 0.24; verb-conjunction/sentence: 0.35.
- **Domain vocabulary**: Top content words (*European, Parliament, Commission, Member States, Council*) provide domain-specific signal opportunities.

Source corpus. The training data originates from the ParlLawSpeech corpus [29], comprising 574,119 European Parliament plenary speeches (1999–2024) in multiple languages. Since language metadata is not provided, we performed language identification using fastText (`lid.176.ftz`) [30], retaining 291,603 English speeches (50.8%). The 300 training documents include 214 MEP speeches (71%) and 86 non-MEP interventions (29%). Alongside the training set, we extracted from a pool of 140,004 English MEP oral speeches three 200-document subcorpora. They contribute to parameter optimisation via a weighted objective score $= \alpha \cdot \text{BAcc}_{\text{train}} + (1 - \alpha) \cdot \text{BAcc}_{\text{val}}$ with $\alpha = 0.5$, while also serving as held-out evaluation sets:

1. **Recent (2020–2024)**: No length constraints (avg. 248 words).
2. **Daterange-matched**: Same time range as training (1999–2024), no length constraints (avg. 292 words).
3. **Length-matched**: Same date range, filtered to 1,500–4,500 characters (avg. 429 words).

Table 1 summarises corpus statistics. The training set (median 406 words, 16 sentences) is substantially longer than typical corpus documents (full corpus median: 177 words), suggesting that the shared task selected longer speeches providing more embedding opportunities. The shorter Daterange and Recent subsets partly explain their lower detection rates due to fewer tokens for signal accumulation.

Table 1
Source, train and validation corpora statistics. Statistics computed using NLTK tokenization.

Corpus	Docs	Time	Words		Sentences	
			Mean(±SD)	Median [P25–P75]	Mean(±SD)	Median [P25–P75]
Full corpus (all lang.)	574,119	1999–2024	216(±224)	177 [103–240]	8.2(±8.9)	6 [4–10]
English speeches	291,603	1999–2024	259(±273)	200 [111–311]	9.6(±10.7)	7 [4–12]
English MEP oral	140,004	1999–2024	303(±192)	251 [182–381]	11.4(±7.6)	10 [6–14]
Training set	300	1999–2024	432(±196)	406 [243–616]	16.9(±8.3)	16 [10–23]
Val. Daterange	200	1999–2024	292(±165)	245 [182–366]	11.1(±6.8)	9 [7–13]
Val. Length	200	1999–2024	429(±136)	395 [319–485]	16.0(±6.2)	15 [11–19]
Val. Recent	200	2020–2024	248(±128)	220 [188–271]	11.2(±6.8)	10 [8–13]

Full English corpus analysis. For v2, we recomputed syntactic baselines on all 291,603 English speeches (MEP oral 140,004; MEP written 98,454; non-MEP 53,145). Table 2 reports the corpus-level baselines. Semicolon rate increases slightly (0.024 → 0.032), passive ratio rises from 0.204 to 0.238, while ACL ratio and verb-conjunction density change marginally.

4. System Overview

The system consists of two components: an *embedder* and a *detector*. The embedder takes a plain text document and produces a watermarked version by applying modifications through three independent

[1] Scripts and results are available at the GitHub repository: https://github.com/davidecolla/pan-clef-2026-wt-jrc-genesis#.

Table 2
Syntactic baselines. v1 uses training data only (300 docs); v2 uses all English speeches.

Metric	Train (v1)	All EN (v2)	MEP oral	Daterange	Length	Recent
Semicolons per sentence	0.0235	0.0319	0.0244	0.0283	0.0205	0.0201
Passive sentence ratio	0.2036	0.2380	0.2286	0.2287	0.2354	0.1449
ACL ratio	0.2363	0.2321	0.1879	0.1775	0.1764	0.1576
Verb-conjunction per sent	0.3531	0.3302	0.3455	0.3553	0.3378	0.3363

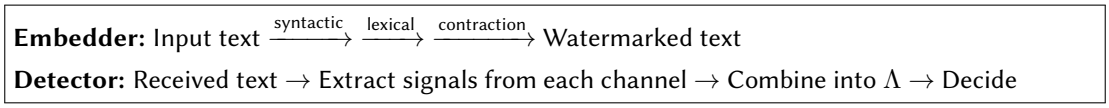

Figure 1: High-level pipeline. The embedder applies three channels sequentially; the detector extracts evidence from each channel independently and combines them into a single decision.

channels. The detector takes a (possibly modified) text and decides whether it contains a watermark. Figure 1 summarises the pipeline.

4.1. Channels

The three channels operate at different linguistic levels:

1. **Lexical channel**: Replaces content words (adjectives, adverbs, nouns, verbs) with synonyms chosen according to a keyed green/red list.
2. **Contraction channel**: Selects between contracted and expanded forms (e.g., "can't" vs. "cannot") based on a keyed bit assignment.
3. **Syntactic channel**: Applies structural transformations (semicolon substitution, clause expansion, voice conversion) that alter syntactic statistics.

The channels are designed to be *orthogonal* to increase its robustness against attacks.

4.2. Shared Secret Key

Lexical and contraction channels derive their decisions from a shared secret key k via HMAC-SHA256. For a given word w at a specific position in the text, we define a *site identifier* s that encodes the token's positional index and a channel-specific prefix. The *bit assignment function* is as follows:

$$b(w, s) = \text{HMAC-SHA256}(k,\ w \| s) \mod 2 \tag{1}$$

where w is the lowercase lemma of the token, s is the site identifier, k is the secret key, and $\|$ denotes string concatenation. The output $b = 1$ denotes a "green" (watermark-carrying) choice and $b = 0$ denotes "red."

4.3. Embedding

The embedder processes the input text sequentially through the three channels. The order matters: syntactic transforms are applied first (since they change sentence structure and token count), followed by lexical substitution (token-index sensitive), and finally contraction manipulation.

4.3.1. Lexical Channel

The lexical channel identifies substitution sites – tokens of part-of-speech ADJ, ADV, NOUN, or VERB that have valid synonyms in WordNet. For each site, the embedder selects a synonym whose bit assignment under the secret key is $b = 1$ (green). Candidate synonyms are filtered through:

1. **Corpus vocabulary filter**: Only synonyms attested in the training corpus (European Parliament proceedings) are considered in v1. The full English partition of the source corpus is used to build the vocabulary in v2. The choice of using only synonyms attested in a reference corpus is made to ensure domain appropriateness.
2. **Synonym selection**:
   - **v1** – Synonyms with Wu–Palmer similarity exceeding a minimum threshold (0.50 for ADJ/ADV, 0.30–0.40 for NOUN/VERB) are chosen.
   - **v2** – Adds BERTScore-based selection: after Wu–Palmer pre-filtering, candidates are scored by computing BERTScore F1 (using DistilBERT-base-uncased) between the original sentence and the sentence with the candidate substitution. Candidates with BERTScore ≥ 0.85 (initial pass) or ≥ 0.80 (refinement rounds) are accepted. Synonym candidates are inflected to match the original word's morphological form (e.g., "adopted" + candidate "embrace" $\rightarrow$ "embraced"), using spaCy morphological features.
3. **Blocklist**: Domain-inappropriate substitutions are explicitly blocked. In v2, abbreviations and punctuation-ending candidates (e.g., "anon.") are also filtered out.
4. **Protected zones**: Named entities, quoted text, and technical terms are never modified.

The embedder performs iterative refinement (up to 8 rounds) to maximise the green rate, targeting $\geq 80\%$ green features.

4.3.2. Contraction Channel

English offers many equivalent forms, e.g. "cannot" vs. "can't", "I am" vs. "I'm". The contraction channel assigns a target bit to each contraction site (identified by canonical form and ordinal position in the text). The embedder selects the contracted form when the target bit is 1, and the expanded form when it is 0.

4.3.3. Syntactic Channel

The syntactic channel applies the following structural transformations:

Verb-conjunction semicolon substitution. When two verbs are coordinated with ", and/but/or", the punctuation and conjunction are replaced with a semicolon. Example: "*I welcome the proposal, and I recall…*" $\rightarrow$ "*I welcome the proposal; I recall…*".

ACL expansion. Past participles attached to nouns via the `acl` dependency are expanded with "that was/were". Example: "*the measures adopted by…*" $\rightarrow$ "*the measures that were adopted by…*".

VBG advcl to relative clause. Present participles functioning as adverbial clauses (without a subordinating conjunction) are converted to relative clauses. Example: "*suggesting that…*" $\rightarrow$ "*which suggests that…*". v2 adds constraints: instrumental verbs (e.g., "using", "employing") are excluded, coordinated participles are skipped, and overlapping transformation sites are filtered to prevent text corruption.

Active to passive voice. Selected active-voice sentences are converted to passive voice. Example: "*The committee approved the proposal*" $\rightarrow$ "*The proposal was approved by the committee*". Both versions apply up to 3 passive transformations per document. v2 applies stricter sentence constraints: sentences with infinitive complements (`xcomp`), clausal complements (`ccomp`), phrasal verbs, comparison verbs, or complex subject/object structures are skipped. v2 also fixes pronoun conversion ("we" $\rightarrow$ "us" in agent phrases) and preserves sentence-ending punctuation.

Institution expansion (v2). Shorthand EU institution references are expanded to their full form (e.g., "the Commission" → "the European Commission") when not already qualified.

These syntactic transforms are not keyed. Instead, they create a *statistical fingerprint*: watermarked texts have more semicolons, fewer participial clauses, and more passive sentences than the corpus baseline.

4.4. Detection

The detector extracts evidence from each channel independently and combines them into a single score Λ. The lexical and contraction channels use a log-likelihood ratio (LLR) test over keyed binary observations. The syntactic channel measures statistical deviations from corpus baselines.

4.4.1. Keyed Channels

Lexical channel. The detector re-identifies all valid substitution sites and checks the bit assignment of each observed word. Let n denote the number of sites found, $g = \sum_{i=1}^{n} x_i$ the count of green observations where $x_i = b(w_i, s_i)$, and $r = n - g$ the count of red observations. Under H_0 (no watermark), each site is green with probability $p_0 = 0.5$. Under H_1 (watermarked), we use $p_1 = 0.70$ (conservative, since the embedder targets 80%). The lexical LLR is:

$$\Lambda_{\text{lex}} = g \cdot \ln \frac{p_1}{p_0} + r \cdot \ln \frac{1 - p_1}{1 - p_0} \tag{2}$$

clamped to $\max(0, \Lambda_{\text{lex}})$.

Contraction channel. The detector locates all contraction/expansion sites and checks whether the observed form matches the keyed target. Let c denote the total sites and m the matching sites. Under H_1, matching probability is $p_1^{\text{ctr}} = 0.95$; under H_0, it is $p_0^{\text{ctr}} = 0.5$. The contraction LLR is:

$$\Lambda_{\text{ctr}} = m \cdot \ln \frac{0.95}{0.5} + (c - m) \cdot \ln \frac{0.05}{0.5} \tag{3}$$

clamped to $\max(0, \Lambda_{\text{ctr}})$.

BERTScore-aligned site identification (v2). The v2 detector uses BERTScore (threshold ≥ 0.80) to validate which tokens are valid substitution sites, matching the embedder's filtering logic. This ensures the detector identifies the exact same word positions the embedder could have modified, improving detection consistency.

4.4.2. Syntactic Channel

Unlike the keyed channels, the syntactic channel measures how much the text's statistics deviate from corpus baselines. System v1 uses four signals; v2 extends this to nine.

Base signals (v1–v2). Four statistical signals measured against corpus baselines: (1) semicolons per sentence, (2) ACL ratio $\text{acl}/(\text{acl} + \text{relcl})$, (3) verb-conjunction count, and (4) passive ratio. Each signal is thresholded and scaled to $[0, 1]$, then combined with weights (semicolons: 0.8, ACL: 0.4, verb-conj: 0.2, passive: 0.4):

$$\sigma_{\text{original}} = \frac{0.8\,\sigma_{\text{semi}} + 0.4\,\sigma_{\text{acl}} + 0.2\,\sigma_{\text{verb}} + 0.4\,\sigma_{\text{pass}}}{1.8} \tag{4}$$

Attack-resistant signals (v2 only). Four additional signals designed to strengthen robustness against attacks: (5) agent phrases per sentence (Cohen's $d = 1.15$), (6) passive subject ratio (Cohen's $d = 0.90$), (7) "that was/were VBN" patterns, and (8) relcl ratio (Cohen's $d = 0.60$). Combined as:

$$\sigma_{\text{new}} = \frac{1.0\,\sigma_{\text{agent}} + 0.8\,\sigma_{\text{nsubj}} + 0.5\,\sigma_{\text{tbv}} + 0.5\,\sigma_{\text{relcl}}}{2.8} \tag{5}$$

The final syntactic signal: $\sigma_{\text{combined}} = (\sigma_{\text{original}} + \sigma_{\text{new}})/2$.

Institution signal (v2 only). Detects elevated "European Parliament/Commission/Council" forms:

$$\sigma_{\text{inst}} = \begin{cases} \min(1.0,\ n_{\text{total}}/10) & \text{if } n_{\text{total}} \geq 3 \text{ and } n_{\text{eu}}/n_{\text{total}} > 0.8 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Aligned passive counting (v2). The passive signal uses `count_transformable_passive_sentences()` which only counts passive sentences matching the embedder's transformation patterns.

4.4.3. Detection Decision

The syntactic and institution signals are converted to LLR-like contributions: $\Lambda_{\text{syn}} = \sigma_{\text{combined}} \times \alpha_{\text{syn}}$ and $\Lambda_{\text{inst}} = \sigma_{\text{inst}} \times \alpha_{\text{inst}}$.

v1 uses a simple threshold on the combined LLR:

$$\Lambda = \Lambda_{\text{lex}} + \Lambda_{\text{ctr}} + \Lambda_{\text{syn}}, \quad \text{detected} = (\Lambda > \tau) \tag{7}$$

with $\tau = 1.50$ and $\alpha_{\text{syn}} = 2.0$.

v2 adds an institution signal and a *gating mechanism* to prevent false positives from noise accumulation:

$$\Lambda = \Lambda_{\text{lex}} + \Lambda_{\text{ctr}} + \Lambda_{\text{syn}} + \Lambda_{\text{inst}} \tag{8}$$

$$\text{detected} = (\Lambda > \tau) \ \wedge \ \left(\text{gate}_{\text{lex/ctr}} \ \vee \ \text{gate}_{\text{syn}}\right) \tag{9}$$

with $\tau = 2.5$, $\alpha_{\text{syn}} = 6.0$, $\alpha_{\text{inst}} = 4.0$, and gate thresholds $\gamma_{\text{lex}} = 3.5$, $\gamma_{\text{ctr}} = 1.0$, $\gamma_{\text{syn}} = 0.25$, $\gamma_{\text{inst}} = 0.2$. The gating condition requires that at least one channel provides independently strong evidence: $\text{gate}_{\text{lex/ctr}}$ fires when $\Lambda_{\text{lex}} > \gamma_{\text{lex}}$ or $\Lambda_{\text{ctr}} > \gamma_{\text{ctr}}$; gate_{syn} fires when $\sigma_{\text{combined}} > \gamma_{\text{syn}}$. This prevents false positives from weak signals across multiple channels accumulating above τ without any single channel being individually convincing. However, this conservative design comes at a cost: under attacks that degrade keyed signals (e.g., L3, L4), watermarked documents may exceed τ yet fail the gate condition, resulting in missed detections (lower TPR) in exchange for high specificity (TNR).

4.4.4. Parameter Optimisation

We employ grid search to optimise detector hyperparameters while ensuring generalisation via a weighted objective balancing training and validation performance ($\alpha = 0.5$). For v1, optimisation sets $\tau = 1.5$ and adjusts signal thresholds (passive: 0.15, semicolons: 0.03, ACL ratio: 0.04), while keeping $\alpha_{\text{syn}} = 2.0$, trading off non-falsifiability for increased detection. For v2, the increased syntactic scale ($\alpha_{\text{syn}} = 6.0$) improves attack resistance (especially in paraphrase attacks), allowing syntactic signals to contribute more heavily when keyed signals are degraded.

5. Independent Evaluation

To assess robustness, we designed an independent evaluation pipeline that applies increasingly aggressive attacks to watermarked text and measures both detection performance and text quality degradation.

5.1. Attack Strategy

Our evaluation applies seven independent attack levels to each text. These attacks are *watermark-agnostic* – motivated by the literature: paraphrase-based evasion [25, 26], lexical substitution attacks [21], human-imperceptible character-level perturbation [31] and combined/partial attacks [32]:

1. **L1 – Punctuation & whitespace normalisation**: Rule-based cleanup of brackets, quotes, spacing, and capitalization.
2. **L2 – Grammar correction**: T5-based grammar error correction.

3. **L3 – Content word substitution**: BERT fill-mask replaces content words with confident predictions.
4. **L4 – Controlled paraphrasing**: T5 paraphrase with BERTScore quality gating.
5. **L5 – Homoglyph substitution**: ASCII characters replaced with visually similar Unicode/Cyrillic characters. Our detector normalises these.
6. **L6 – Mixed attack**: L1 plus a randomly selected L2–L5 attack per sentence.
7. **L7 – Partial attack**: Only 50% of sentences are attacked (using L6 strategy).

Each level receives the *original* watermarked text as input (not the output of previous levels), enabling independent measurement of each attack's impact.

5.2. Evaluation Metrics

Following the PAN 2026 evaluation protocol, we report the **Text Watermarking F-score (TWF)**:

$$\text{TWF} = \max(\text{BLEU}, \text{BERTScore}) \times \text{BAcc} \tag{10}$$

where **BAcc** (Balanced Accuracy) is the mean of TPR and TNR, and **BERTScore (F1)** measures semantic similarity between original and watermarked text using RoBERTa-large embeddings.

5.3. Results and Discussion

Table 3 reports detection performance on non-attacked texts (L0). Both detectors achieve high balanced accuracy (>0.97).

Table 3
Detection performance on non-attacked texts (L0).

		v1			v2		
Dataset	N	TPR	TNR	BAcc	TPR	TNR	BAcc
Train	300	1.000	0.963	0.982	1.000	0.953	0.977
Daterange	200	1.000	0.915	0.958	0.995	0.950	0.972
Length	200	1.000	0.995	0.998	1.000	0.990	0.995
Recent	200	1.000	0.950	0.975	0.995	0.945	0.970
Mean		1.000	0.956	0.978	0.998	0.960	0.979

Table 4 presents detection balanced accuracy under attacks. v2 substantially outperforms v1 on L4 (T5 paraphrase), the most challenging attack, thanks to its higher syntactic scaling ($\alpha_{\text{syn}} = 6.0$) which partially compensates for the destroyed lexical signal. Nonetheless, v2 is less robust against L3 attack (BERT fill-mask for synonym substitution).

Table 5 reports the TWF metric combining text quality and detection accuracy. v2 achieves higher TWF due to both its improved BERTScore (O→W) of 0.975 (vs. 0.953 for v1) and competitive detection accuracy.

In the official test set, v1 achieves a BAcc of 0.720, a BERTScore of 0.952, TPR of 0.226 (FNR of 0.274), TNR of 0.494 (FPR of 0.006), and TWF of **0.686**. v2 a BAcc of 0.785, a BERTScore of 0.969, TPR of 0.299 (FNR of 0.201), TNR of 0.486 (FPR of 0.014), and TWF of **0.761**.

v1 vs. v2. Comparing the two versions on attacked data, both original and watermarked (Table 4), it is evident that they behave comparably on all attack levels except for L2, L3 and L4. v2 has its most substantial gain on L4, where mean BAcc rises from 62.4% to 77.1% (+14.7 pp), and a moderate one in L2, grammar correction (86.2% → 91.6%). This means that v2, does not trade off higher BERTScore for balanced accuracy (v2's BS(O→W) = 0.975 compared to 0.953 for v1). However, v2 shows a substantial decrease on L3 (content word substitution): 89.6% vs. v1's 98.2%, signalling that syntactic signal alone is

Table 4
Balanced Accuracy under attacks (L1–L7).

Dataset	Detector	L1	L2	L3	L4	L5	L6	L7	Avg
Train	v1	0.987	0.902	0.987	0.677	0.987	0.877	0.942	0.908
	v2	0.975	0.948	0.915	0.800	0.977	0.913	0.955	0.926
Daterange	v1	0.968	0.833	0.968	0.593	0.968	0.855	0.920	0.872
	v2	0.972	0.900	0.880	0.755	0.972	0.903	0.935	0.902
Length	v1	0.995	0.905	0.998	0.605	0.998	0.877	0.968	0.906
	v2	0.995	0.958	0.943	0.820	0.995	0.953	0.978	0.949
Recent	v1	0.975	0.810	0.978	0.620	0.978	0.825	0.917	0.872
	v2	0.965	0.860	0.847	0.708	0.968	0.823	0.913	0.869
Mean	v1	0.981	0.862	0.982	0.624	0.982	0.859	0.937	0.890
	v2	0.977	0.916	0.896	0.771	0.978	0.898	0.945	0.912

Table 5
Text Watermarking F-score (TWF = BERTScore(O→W) × BAcc).

Dataset	Detector	L1	L2	L3	L4	L5	L6	L7	Avg
Train	v1	0.941	0.860	0.941	0.645	0.941	0.836	0.898	0.866
	v2	0.954	0.916	0.880	0.751	0.956	0.877	0.926	0.894
Daterange	v1	0.923	0.794	0.923	0.565	0.923	0.815	0.877	0.831
	v2	0.939	0.858	0.848	0.695	0.934	0.866	0.905	0.864
Length	v1	0.948	0.862	0.951	0.577	0.951	0.836	0.923	0.864
	v2	0.963	0.936	0.907	0.770	0.963	0.907	0.941	0.912
Recent	v1	0.929	0.772	0.932	0.591	0.932	0.786	0.874	0.831
	v2	0.941	0.839	0.809	0.681	0.946	0.775	0.868	0.837
Mean	v1	0.935	0.822	0.937	0.594	0.937	0.818	0.893	**0.848**
	v2	0.949	0.887	0.861	0.724	0.950	0.856	0.910	**0.877**

not strong enough for accurate detection in v2 setting (higher syntactic signal and gating mechanism, preventing false positives, but obscuring degraded signal in true positives).

Statistical significance. To substantiate the the differences between v1 and v2, we applied McNemar's paired test on per-document detection outcomes across all datasets and attack levels ($N = 12{,}600$ paired observations), considering only the watermarked partition.[2] The overall improvement is statistically significant ($\chi^2 = 6.44$, $p = 0.011$), with v2 correctly classifying 733 documents that v1 missed, while v1 was correct on only 639 documents where v2 failed (Table 6). v2 statistically significantly outperforms v1 in two datasets (Daterange and Length). The Recent dataset is the only one where v1 overall outperforms v2 (88.2% vs. 86.4%, $p < 0.01$). The two versions perform comparably in the Train set.

Per-attack analysis (Table 7) reveals that v2's improvement concentrates on L4 (T5 paraphrase), where v2 achieves 76.3% vs. v1's 66.8% – a +9.5 pp gain ($\chi^2 = 67.84$, $p < 0.001$). This gain is attributable to v2's higher syntactic scaling ($\alpha_{\text{syn}} = 6.0$) and additional syntactic signals that partially compensate for the lexical signal destroyed by neural paraphrasing. L2 (grammar correction, +2.6 pp, $p = 0.002$) also shows a significant improvement. On attacks that preserve token-level signals (L1 punctuation, L5 homoglyph), the two systems perform identically (≈97.7%, $p > 0.5$), confirming that v2's design changes do not degrade performance on benign attack types. v1 significantly outperforms v2 on L3 (content word substitution): 97.6% vs. 89.6% ($\chi^2 = 103.68$, $p < 0.001$). This is the strongest per-level effect in the analysis. L3 specifically targets content words and v2's higher detection threshold ($\tau = 2.5$

[2]Please notice that in Table 4 the mean BAcc is computed on original too.

Table 6
McNemar's test for v1 vs. v2 (aggregate across attack levels) on watermarked data. Statistically significant results are marked with an asterisk.

Dataset	N	v1 Acc	v2 Acc	v1>v2	v2>v1	p
Train	4200	0.917	0.922	218	240	0.304
Daterange	2800	0.880	0.895	143	186	0.018*
Length	2800	0.915	0.944	69	150	<0.001*
Recent	2800	0.882	0.864	209	157	<0.01*
Overall	12600	0.900	0.908	639	733	0.011*

vs. $\tau = 1.5$) plus the gating mechanism require more accumulated evidence that is destroyed by this targeted substitution. L6 shows a consistent trend in v2's favour but does not reach significance at the aggregate level ($p = 0.092$). L7 (partial attack) shows no significant difference ($p = 0.80$), suggesting both systems reach a comparable floor with this attack type.

Table 7
McNemar's test for v1 vs. v2 per attack level (aggregate across all watermarked datasets). Asterisks mark statistically significant results.

Level	Attack	N	v1 Acc	v2 Acc	v1>v2	v2>v1	p
L1	Punctuation	1800	0.977	0.976	29	28	0.895
L2	Grammar	1800	0.887	0.913	89	135	0.002*
L3	Content word sub.	1800	0.976	0.896	172	28	<0.001*
L4	T5 paraphrase	1800	0.668	0.763	130	301	<0.001*
L5	Homoglyph	1800	0.978	0.977	31	29	0.796
L6	Mixed	1800	0.877	0.892	115	142	0.092
L7	Partial attack	1800	0.941	0.939	73	70	0.802

In summary, the aggregate improvement from v1 to v2 is statistically significant ($p = 0.011$) but modest in magnitude (+0.8 pp). The two systems exhibit complementary strengths: v2 excels under paraphrasing attacks (L4: +9.5 pp) thanks to its redundant syntactic signals that partially compensate when keyed signals are destroyed, while v1 retains an advantage under content word substitution (L3: +8.0 pp) due to its lower threshold and more aggressive watermarking (lower BERTScore). The Length-matched dataset, with its longer documents providing more evidence for both channels, shows the clearest v2 advantage ($p < 0.001$, +2.9 pp).

Ablation: multi-channel vs. single-channel. To verify the contribution of the individual channels against the combined system robustness, we performed an ablation study comparing the full v1 detector against each channel operating in isolation (using the same LLR threshold $\tau = 1.5$).[3] Table 8 reports McNemar's test results aggregated across all datasets and attack levels ($N = 12{,}600$).

Table 8
Ablation (v1): combined vs. single-channel detectors. All differences are highly significant ($p < 0.001$, McNemar's test).

Detector	Accuracy	Δ	**Combined** over single-channel	**Single-channel** over combined	χ^2
Combined	0.900	—	—	—	—
Lexical-only	0.796	−0.104	1483	170	1042.9
Contraction-only	0.779	−0.121	1727	202	1205.6
Syntactic-only	0.511	−0.390	5159	251	4452.6

[3] We carried out the tests only on v1 as it is lighter than v2.

The combined system outperforms every individual channel with high significance ($p < 0.001$). Table 8 quantitatively confirms the design rationale: the combination provides robustness that no individual channel achieves alone.

6. Conclusion

We presented a multi-channel linguistic watermarking system that combines keyed synonym substitution, keyed contraction manipulation, and corpus-based syntactic transforms into a unified LLR-based detection framework. We submitted two versions: v1 provides the baseline three-channel system; v2 extends this with BERTScore-based synonym selection (improving text quality from 0.953 to 0.975 BERTScore), and stricter syntactic signal transformations.

The multi-channel design provides robustness, as no single-channel outperforms it. The system is lightweight, requiring no neural inference at detection time for v1 and minimal inference for v2's site validation.

Limitations include sensitivity to heavy paraphrasing that simultaneously replaces vocabulary and restructures syntax, and domain-specificity to English parliamentary text. Future work could explore coarse-grained keyed syntactic fingerprints that balance non-falsifiability with attack resistance.

Acknowledgments

We want to thank the University of Turin for granting access to the Sketch Engine tool used for corpus analysis.

Declaration on Generative AI

During the preparation of this work, the authors used Kiro (Claude Opus 4.5) for: code generation, implementation assistance, and drafting of this notebook paper. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. Bevendorff, M. Fröbe, A. Greiner-Petter, A. Jakoby, M. Mayerl, P. Nakov, H. Plutz, M. Potthast, B. Stein, M. N. Ta, Y. Wang, E. Zangerle, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[2] H. Plutz, A. Jakoby, J. Bevendorff, M. Potthast, B. Stein, Overview of the Text Watermarking Task at PAN 2026, in: A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

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

[4] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, A watermark for large language models, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), Proceedings of the 40th International Conference on Machine Learning, volume 202 of *Proceedings of*

Machine Learning Research, PMLR, 2023, pp. 17061–17084. URL: https://proceedings.mlr.press/v202/kirchenbauer23a.html.

[5] J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman, Electronic marking and identification techniques to discourage document copying, IEEE Journal on Selected Areas in Communications 13 (1995) 1495–1504. doi:10.1109/49.464718.

[6] N. F. Maxemchuk, Electronic document distribution, AT&T technical journal 73 (1994) 73–80.

[7] S. Low, N. Maxemchuk, Performance comparison of two text marking methods, IEEE Journal on Selected Areas in Communications 16 (1998) 561–572. doi:10.1109/49.668978.

[8] D. Zou, Y. Shi, Formatted text document data hiding robust to printing, copying and scanning, in: 2005 IEEE International Symposium on Circuits and Systems (ISCAS), 2005, pp. 4971–4974 Vol. 5. doi:10.1109/ISCAS.2005.1465749.

[9] D. Huang, H. Yan, Interword distance changes represented by sine waves for watermarking text images, IEEE Transactions on Circuits and Systems for Video Technology 11 (2001) 1237–1245. doi:10.1109/76.974678.

[10] A. M. Alattar, O. M. Alattar, Watermarking electronic text documents containing justified paragraphs and irregular line spacing, in: Security, Steganography, and Watermarking of Multimedia Contents VI, volume 5306, SPIE, 2004, pp. 685–695.

[11] C. Culnane, H. Treharne, A. T. S. Ho, A new multi-set modulation technique for increasing hiding capacity of binary watermark for print and scan processes, in: Y. Q. Shi, B. Jeon (Eds.), Digital Watermarking, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 96–110.

[12] L. Y. Por, K. Wong, K. O. Chee, Unispach: A text-based data hiding method using unicode space characters, Journal of Systems and Software 85 (2012) 1075–1082. URL: https://www.sciencedirect.com/science/article/pii/S0164121211003177. doi:https://doi.org/10.1016/j.jss.2011.12.023.

[13] S. G. Rizzo, F. Bertini, D. Montesi, Content-preserving text watermarking through unicode homoglyph substitution, in: Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 97–104. URL: https://doi.org/10.1145/2938503.2938510. doi:10.1145/2938503.2938510.

[14] U. Topkara, M. Topkara, M. J. Atallah, The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions, in: Proceedings of the 8th Workshop on Multimedia and Security, MM&Sec '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 164–174. URL: https://doi.org/10.1145/1161366.1161397. doi:10.1145/1161366.1161397.

[15] C. Fellbaum, WordNet: An electronic lexical database, MIT press, 1998.

[16] M. J. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, S. Naik, Natural language watermarking: Design, analysis, and a proof-of-concept implementation, in: I. S. Moskowitz (Ed.), Information Hiding, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 185–200.

[17] H. M. Meral, E. Sevinç, E. Ünkar, B. Sankur, A. S. Özsoy, T. Güngör, Syntactic tools for text watermarking, in: E. J. D. III, P. W. Wong (Eds.), Security, Steganography, and Watermarking of Multimedia Contents IX, volume 6505, International Society for Optics and Photonics, SPIE, 2007, p. 65050X. URL: https://doi.org/10.1117/12.708111. doi:10.1117/12.708111.

[18] H. M. Meral, B. Sankur, A. Sumru Özsoy, T. Güngör, E. Sevinç, Natural language watermarking via morphosyntactic alterations, Computer Speech & Language 23 (2009) 107–125. URL: https://www.sciencedirect.com/science/article/pii/S0885230808000284. doi:https://doi.org/10.1016/j.csl.2008.04.001.

[19] Z. Wu, M. Palmer, Verb semantics and lexical selection, in: 32nd annual meeting of the association for computational linguistics, 1994, pp. 133–138.

[20] M. Topkara, C. M. Taskiran, E. J. D. III, Natural language watermarking, in: E. J. D. III, P. W. Wong (Eds.), Security, Steganography, and Watermarking of Multimedia Contents VII, volume 5681, International Society for Optics and Photonics, SPIE, 2005, pp. 441 – 452. URL: https://doi.org/10.1117/12.593790. doi:10.1117/12.593790.

[21] J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K. Fernando, A. Saha, M. Goldblum, T. Goldstein, On the reliability of watermarks for large language models, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=DEJIDCmWOz.

[22] M. Christ, S. Gunn, O. Zamir, Undetectable watermarks for language models, in: The Thirty Seventh Annual Conference on Learning Theory, PMLR, 2024, pp. 1125–1139.

[23] S. Dathathri, A. See, S. Ghaisas, P.-S. Huang, R. McAdam, J. Welbl, V. Bachani, A. Kaskasoli, R. Stanforth, T. Matejovicova, J. Hayes, N. Vyas, M. Al Merey, J. Brown-Cohen, R. Bunel, B. Balle, T. Cemgil, Z. Ahmed, K. Stacpoole, I. Shumailov, C. Baetu, S. Gowal, D. Hassabis, P. Kohli, Scalable watermarking for identifying large language model outputs, Nature 634 (2024) 818–823. URL: https://doi.org/10.1038/s41586-024-08025-4. doi:10.1038/s41586-024-08025-4.

[24] X. Zhao, P. V. Ananth, L. Li, Y.-X. Wang, Provable robust watermarking for AI-generated text, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=SsmT8aO45L.

[25] K. Krishna, Y. Song, M. Karpinska, J. Wieting, M. Iyyer, Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 27469–27500. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/575c450013d0e99e4b0ecf82bd1afaa4-Paper-Conference.pdf.

[26] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, S. Feizi, Can AI-generated text be reliably detected?, 2024. URL: https://openreview.net/forum?id=NvSwR4IvLO.

[27] A. Kilgarriff, V. Baisa, J. Bušta, M. Jakubíček, V. Kovář, J. Michelfeit, P. Rychlỳ, V. Suchomel, The sketch engine, Lexicography 1 (2014) 7–36.

[28] M. A. Covington, J. D. McFall, Cutting the gordian knot: The moving-average type–token ratio (mattr), Journal of quantitative linguistics 17 (2010) 94–100.

[29] J. Schwalbach, L. Hetzer, S.-O. Proksch, C. Rauh, M. Sebok, Parllawspeech, GESIS, Cologne (2025).

[30] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: M. Lapata, P. Blunsom, A. Koller (Eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 427–431. URL: https://aclanthology.org/E17-2068/.

[31] N. Boucher, I. Shumailov, R. Anderson, N. Papernot, Bad characters: Imperceptible nlp attacks, in: 2022 IEEE symposium on security and privacy (SP), IEEE, 2022, pp. 1987–2004.

[32] Q. Pang, S. Hu, W. Zheng, V. Smith, Attacking llm watermarks by exploiting their strengths, in: ICLR 2024 Workshop on Secure and Trustworthy Large Language Models, 2024.